REJ10J0886-0100Z

Everywhere you imagine. RENESAS

High-performance Embedded Workshop V.4.00

User's Manual

Renesas Microcomputer Development Environment System

User's Manual

Rev.1.00
Jan. 19, 2005

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

Introduction

The High-performance Embedded Workshop is a powerful development environment for embedded applications targeted at Renesas micro-controllers. The main features are:

- A configurable build engine that allows you to set-up compiler, assembler and linker options by using GUI.
- An integrated text editor with user customizable syntax coloring to improve code readability.
- A configurable environment, which allows you to run your own tools.
- An integrated debugger, which allows you to build and debug in the same application.
- Version control support.

The High-performance Embedded Workshop has been designed with two key aims; firstly to provide you, the user, with a set of powerful development tools and, secondly, to unify and present them in a way that is easy to use.

About This Manual

This manual describes the High-performance Embedded Workshop system. This manual describes information on the basic "look and feel" of the High-performance Embedded Workshop and customizing the High-performance Embedded Workshop environment and detail the build and the debugging functions common to the High-performance Embedded Workshop products. Figure in the High-performance Embedded Workshop part are those of the SH series. For detailed information about debugging platform, refer to the separate Debugging Platform User's Manual.

This manual does not intend to explain how to write C/C++ or assembly language programs, how to use any particular operating system or how best to tailor code for the individual devices. These issues are left to the respective manuals.

Note:

The High-performance Embedded Workshop does not run on Windows® 3.1 and Windows® 95.

Document Conventions

This manual uses the following typographic conventions:

Convention	Meaning
[Menu->Menu Option]	'->' is used to indicate menu options (for example, [File->Save As...]).
FILENAME.C	Uppercase names are used to indicate filenames.
"enter this string"	Used to indicate text that must be entered (excluding the "" quotes).
Key + Key	Used to indicate required key presses. For example, CTRL+N means press the CTRL key and then, whilst holding the CTRL key down, press the N key.

Trademarks

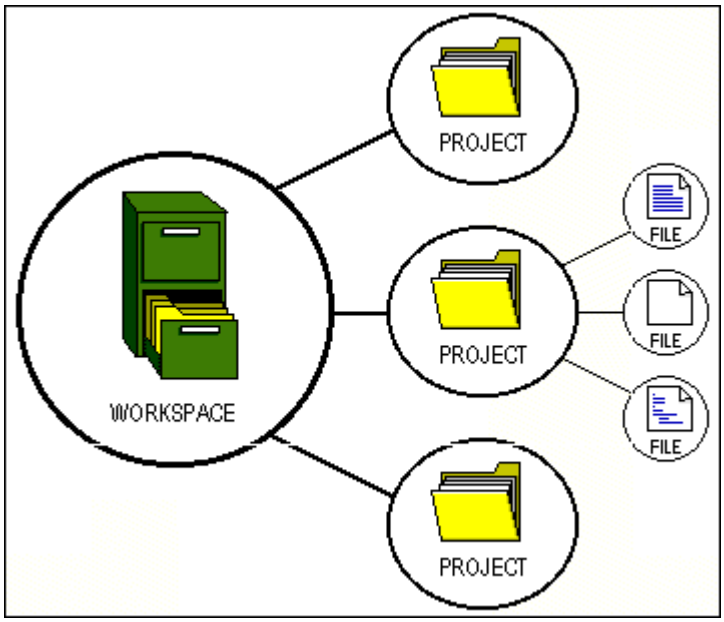
Table Of Contents

1. Overview

The functions for HEW version V.4.00.00 are explained in this manual. This chapter describes the fundamental concepts of the High-performance Embedded Workshop (HEW).

1.1 Workspaces, projects and files

Just as a word processor allows you to create and modify documents, the High-performance Embedded Workshop allows you to create and modify workspaces. A workspace can be thought of as a container of projects and, similarly, a project can be thought of as a container of project files. Thus, each workspace contains one or more projects and each project contains one or more files. The figure below illustrates this graphically:

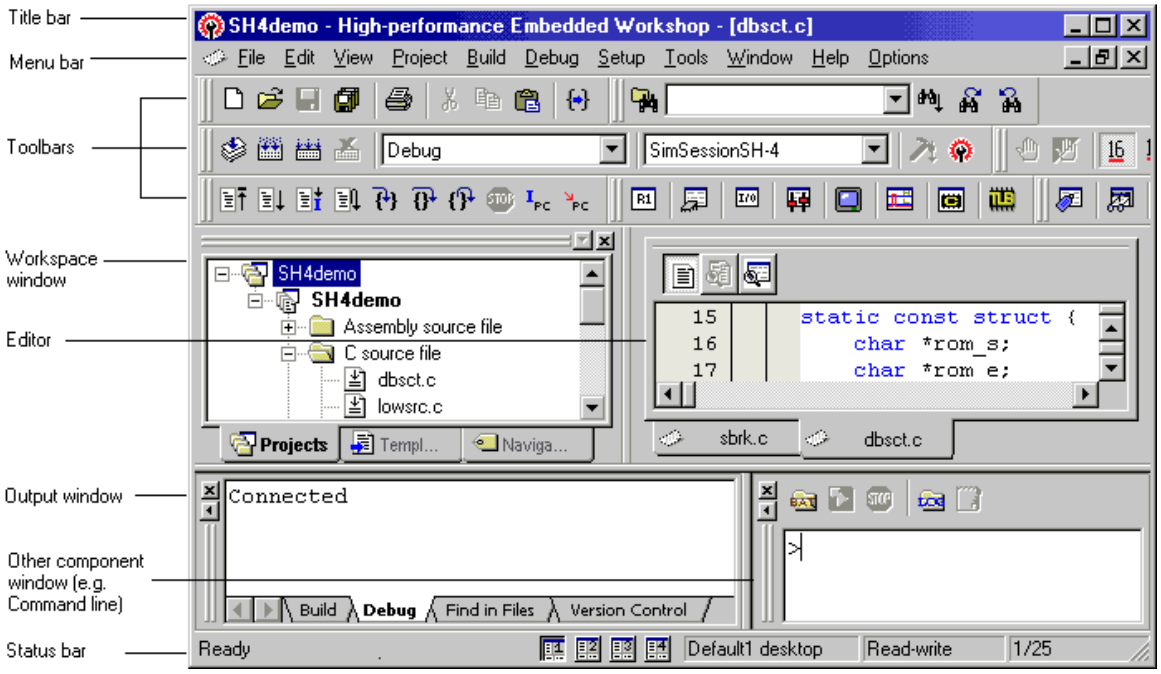


Workspaces allow you to group related projects together. For example, you may have an application that needs to be built for different processors, or you may be developing an application and library at the same time. Projects can also be linked hierarchically within a workspace, which means that when one project is built all of its 'child' projects are built first.

However, workspaces on their own are not very useful – we need to add a project to a workspace and then add files to that project before we can actually do anything.

1.2 Main window

There are three main windows: the **Workspace** window, the **Editor** window and the **Output** window. The **Workspace** window shows the projects and files that are currently in the workspace; the **Editor** window provides file viewing and editing facilities; and the **Output** window shows the results of a various processes (e.g. build, version control commands and so on).



1.2.1 Title bar

The title bar displays the name of current activate project and file. It also contains the standard **Minimize**, **Maximize** and **Close** buttons. Click the **Minimize** button to minimize the HEW on the Windows® task bar. Click the **Maximize** button to force HEW to fill the screen. Click the **Close** button to close the HEW (this has the same effect as selecting the [File→Exit] menu option, or pressing ALT+F4).

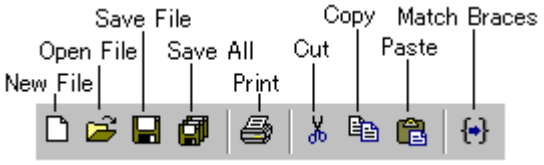
1.2.2 Menu bar

The menu bar initially contains ten menus: **File, Edit, View, Project, Build, Debug, Setup, Tools, Window** and **Help**. All of the menu options are grouped logically under these headings. For instance, if you want to open a file then the **File** menu is where you will find the right menu option; if you want to set-up a tool then the **Tools** menu is the correct selection.

1.2.3 Toolbars

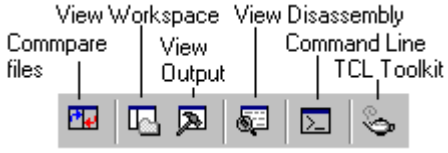
The toolbars provide a shortcut to the options that you will use the most often. There are ten default toolbars: [Editor], [Standard], [Default Window], [Search], [Bookmarks], [Templates], [Debug], [Debug Run], [Map] and [Version Control] (as shown in the figures below). In the initial display of the default session, buttons for version management, and peripheral function are not displayed on the toolbar. Toolbars can be created, modified and removed via the [Tools→Customize] menu option (see the chapter 6, “Customizing the Environment”, for further information).

Editor toolbar:



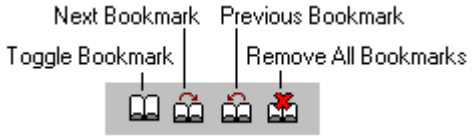
Standard toolbar:

Default Window toolbar:



Search toolbar:

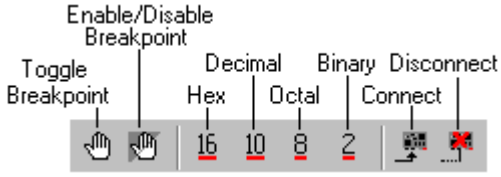
Bookmarks toolbar:



Templates toolbar:

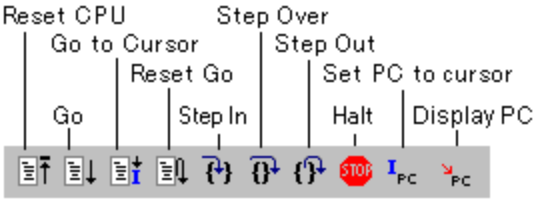
Debug toolbar:

This toolbar is only available when a session is being used which has a target attached.



Debug Run toolbar:

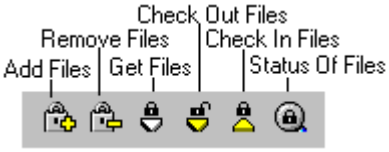
This toolbar is only available when a session is being used which has a target attached.



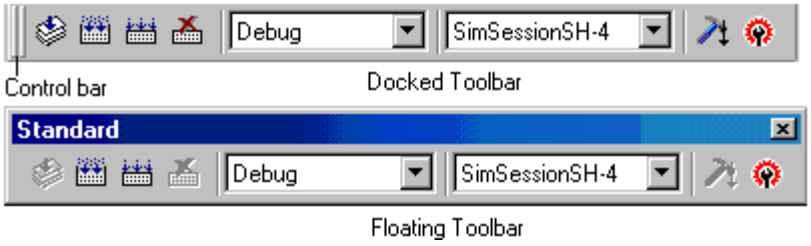
Map toolbar:

Version Control toolbar:

This toolbar is only available when a version control tool is being used in the current project.



When the Standard toolbar is docked, it has a Control bar as shown in the figure below. If you want to move the docked Standard toolbar, click and drag its Control bar to the new location. The figure below shows the Standard toolbar when it is docked and also when it is floating.



To dock a toolbar:

Select one of the following operations:

- Double-click on the title bar of a floating toolbar, **OR**
- Drag the title bar of a floating toolbar and draw it toward an edge of a docked window, menu bar, toolbar or the HEW main frame, on whose edge you would like to dock the window, until the shape of the floating bar changes.

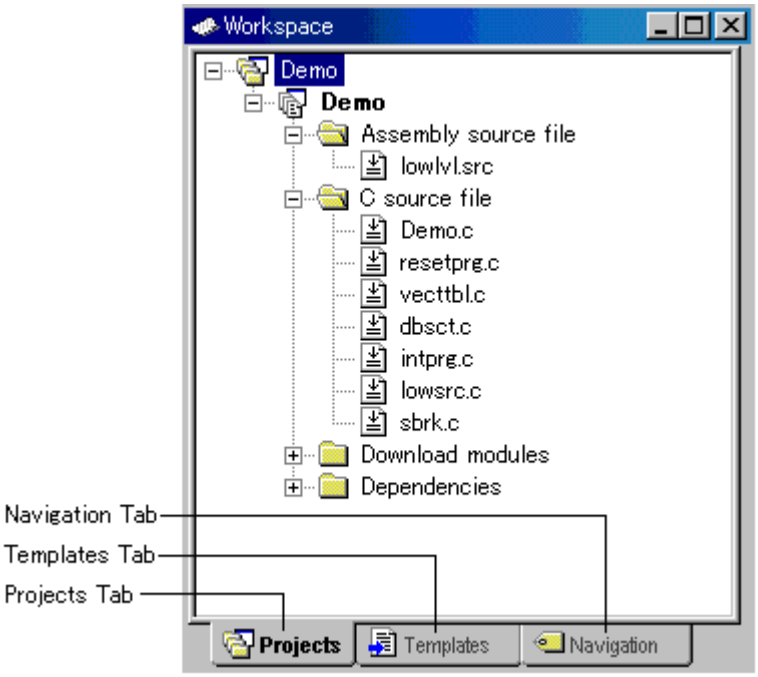
To float a toolbar:

Select one of the following operations:

- Double-click on the control bar of a docked toolbar, **OR**
- Drag the control bar of a docked toolbar and draw it away from the edge of the HEW main frame and from an edge of the other docked windows, menu bar or toolbars.

1.2.4 Workspace window

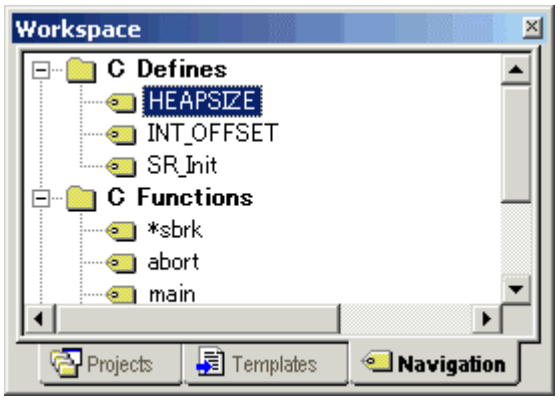
The **Workspace** window has three panes.



The **Projects** tab shows the current workspace, projects and files. You can quickly open any project file or dependent file by double-clicking on its icon. See "2.2 Configuring the workspace window ([Projects] tab)", for more information on the "Workspace" window.

The **Templates** tab displays template settings. See 4.13, "Templates", for more information about a template.

The **Navigation** tab provides jumps to various textual constructs within your project's files. What is actually displayed within the **Navigation** tab depends upon what components are currently installed. The figure below shows ANSI C functions. See chapter 12, "Navigation facilities", for more information about a navigation.

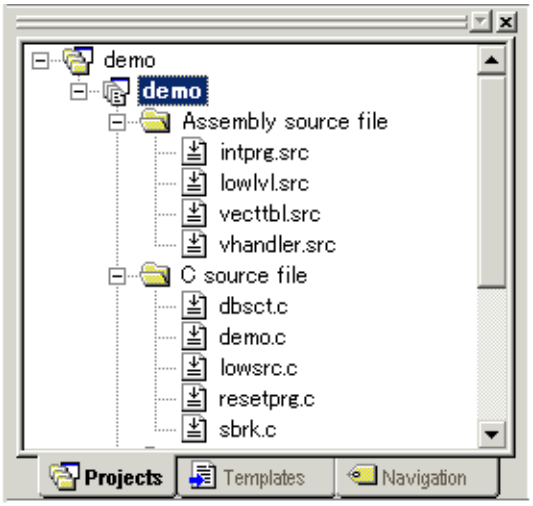


To allow the Workspace window or the Output window docking:

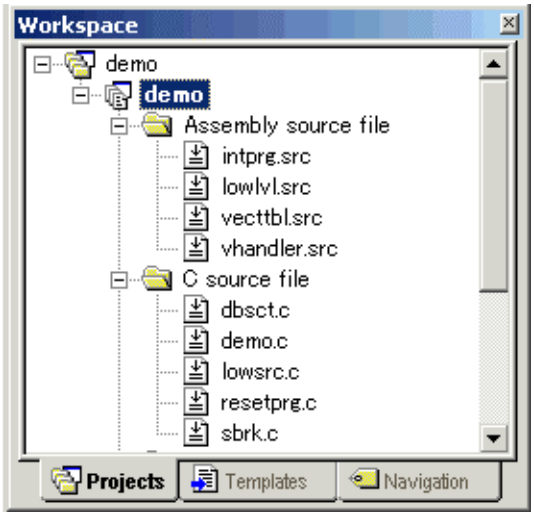
Right-click anywhere inside the **Workspace** window or the **Output** window. Then a pop-up menu will be displayed. If the **Allow Docking** option is checked, docking is allowed. Otherwise, docking is not allowed. Select the **Allow Docking** option to check or uncheck it.

When the **Allow Docking** option is checked, you can dock a window, toolbar or menu bar to the edge of the HEW main window or to the edge of another docked window. You can also float them 'above' the other HEW windows or outside the HEW main window. Figure (i) below shows a docked "Workspace" window, and figure (ii) below shows a floating "Workspace" window.

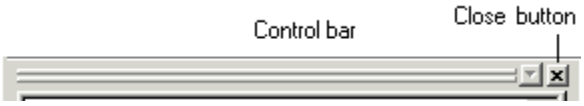
(i)



(ii)



When the **Workspace** window or the **Output** window is docked, it has a control bar as shown below. If you want to move a docked window, click and drag its control bar to the new location.



To dock the Workspace window or the Output window:

1. Ensure that the **Allow Docking** option is checked on the window's pop-up menu.
2. Select one of the following operations:
   - Double-click on the title bar of a floating window, **OR**
   - Drag the title bar of a floating window and draw it toward an edge of a docked window, menu bar or toolbar, or the HEW main frame, on whose edge you would like to dock the windows.

To float the Workspace window or the Output window:

1. Ensure that the **Allow Docking** option is checked on the window's pop-up menu.
2. Select one of the following operations:
   - Double-click on the control bar of a docked window, **OR**
   - Drag the control bar of a docked window and draw it away from the edge of the HEW main frame and from an edge of the other docked windows, menu bar or toolbar.
   - Drag the control bar of a docked window while pressing the "Ctrl" key.

To hide the Workspace window or the Output window:

Select one of the following operations:

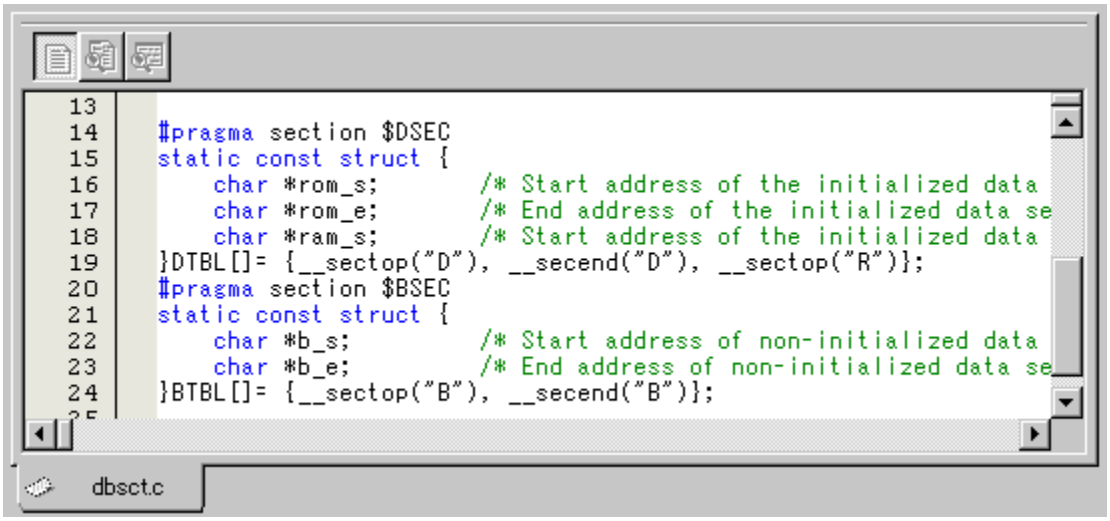
- Click on the close button, which is located in the top right corner of the window, **OR**
- Right-click anywhere inside a floating window and select the **Hide** option on the pop-up menu.

To show the Workspace window or the Output window:

Select the [View->Workspace] or [View->Output] menu option respectively.

1.2.5 Editor window

The editor window is where you will work with the files of your project. The HEW allows you to have many files open at one time, to switch between them, to arrange them and to edit them in whichever order you want to. By default, the editor window is displayed in a notebook style. This means that each file has a separate tab associated with it to aid in navigating between files (see the figure below).



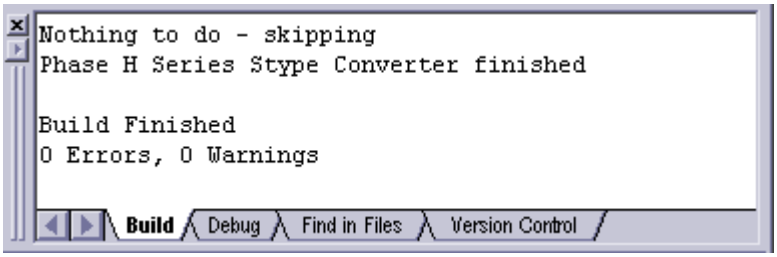
The editor contains a gutter on the left-hand side of the window. The standard column allows the user to configure the position of bookmarks and software breakpoints quickly and easily. If you are unsure what purpose a column has or what the information it is displaying is if you place the cursor over the column a tool tip is displayed showing its identity.

The editor window can be customized via the [Format Views] dialog box, which can be invoked via the [Setup->Format Views…] menu option. This dialog allows you to configure fonts, colors, tabs and so on for the editor window. It also allows the user to change the look of other views, which have been installed by HEW. If you would prefer to use your favorite editor rather than the HEW internal editor then specify your alternative in the [Options] dialog box, which can be invoked via the [Setup->Options…] menu option. For further details on how to use the editor, see chapter 4, Using the Editor.

1.2.6 Output window

To show the **Output** window, select the [View→Output] menu option.

The **Output** window, by default, has four tabs on display.



Use the pop-up menu to clear or save the content of the window.

The Build tab

The **Build** tab shows the output from any build process (e.g. compiler, assembler and so on). If an error is encountered in a source file then the error will be displayed in the build tab, along with the source file name and line number. To quickly locate a problem, double click on the error to jump to the source file and line.

The Debug tab

The **Debug** tab shows the output from any debugger process. Any debug component that needs to display information will send its output to this window.

The Find in Files tab

The **Find in Files** tab displays the results of the last **Find in Files** action. To activate find in files, select the [Edit→Find in Files] menu option, or click the Find In Files toolbar button. For further details on using **Find in Files**, see section 4.6.2, Finding text in multiple files.

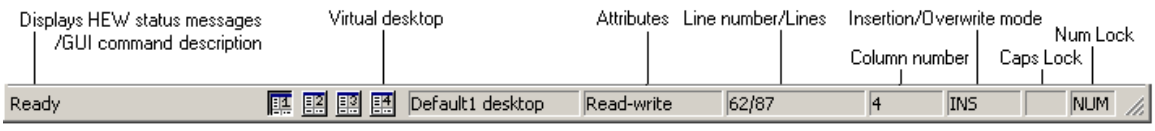
The Version Control tab

The **Version Control** tab displays the results of version control actions. The tab is only displayed if a version control system is in use. For further details on version control, see chapter 7, Version control.

Press the "Shift+Esc" key, and the **Output** window closes.

1.2.7 Status bar

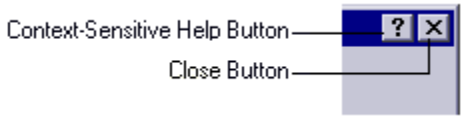
The status bar displays information about the current state of the HEW. The figure below shows the status bar.



1.3 Help system

The help menu is the rightmost menu on the High-performance Embedded Workshop menu bar. It contains the [Contents] menu option, which, when selected, takes you to the main High-performance Embedded Workshop help window.

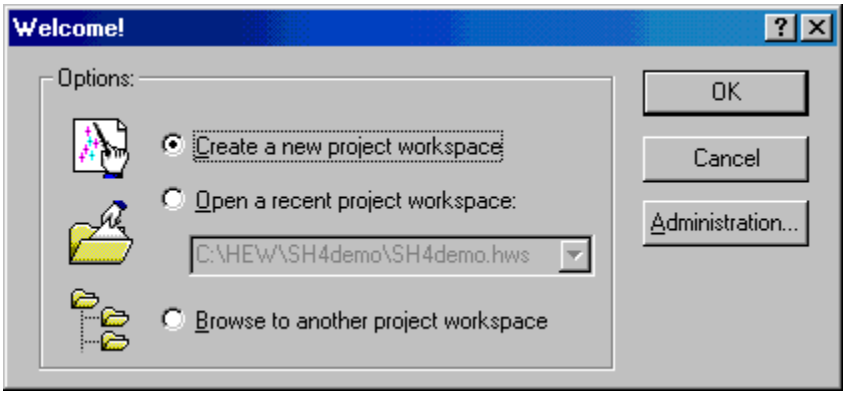
To obtain help on specific dialogs click the context sensitive help button, which is located in the top right-hand corner of each dialog (as shown in the figure below).



When this is clicked, the mouse pointer will change to a pointer with a question mark above it. Whilst the mouse pointer is in this state, click on the part of the dialog that you require assistance on. Alternatively, select the control for which you require help, and press the F1 key.

1.4 Launching the HEW

To initiate the HEW, open the [Start] menu of Windows®, select [Programs], select [Renesas], select [High-performance Embedded Workshop], and then select the shortcut of the HEW. The [Welcome!] dialog box shown will be displayed by default.



To create a new workspace, select the [Create a new project workspace] button, and click the [OK] button. To open one of the recent project workspaces, select the [Open a recent project workspace] button, select a workspace from the drop-down list, and click the [OK] button. The [Recent project workspace] list displays the same content as that seen in the workspace most recently used file list. This list also appears on the file menu. To open a workspace by specifying a workspace file (.HWS file), select the [Browse to another project workspace] button, and click the [OK] button. To register or unregister a tool from the HEW, click the [Administration] button. Click the [Cancel] button to use the HEW without opening a workspace.

1.5 Creating a new workspace

To create a new workspace:

1. Select the [Create a new project workspace] option from the [Welcome!] dialog box and click the [OK] button or select [File->New Workspace...]. The [New Project Workspace] dialog box will be displayed.
2. Enter the name of the new workspace into the [Workspace Name] field. This can be up to 32 characters in length and contain letters, numbers, and the underscore character. Especially, do not use a minus sign, or a space. As you enter the workspace name, the HEW will add a sub-directory and project name for you automatically. This can be changed if desired. This allows the workspace and project name to be different. To select the directory in which you would like to create the workspace, use the [Browse…] button or type the directory into the [Directory] field manually.
3. Select the [CPU family] and [Toolchain] upon which you would like to base the workspace. Note that these cannot be changed once the workspace has been created.
4. When a new workspace is created, the HEW will also automatically create a project with the name specified in the [Project Name] field and place it inside the new workspace. The project types list displays all of the available project types (e.g. Application, Library etc.). Select the type of project that you want to create from this list. The project types displayed will be all valid types for the current pair of [CPU family] and [Toolchain]. The project types are classified in three classes: toolchain-only, debugger-only, and full project generator that configures both the debugger and toolchain aspect of the HEW.
5. Click the [OK] button to create the new workspace and project. This then launches the wizard you have selected to guide you through the creation process.

Note:

It is not possible to create a workspace if one already exists in the same directory.

1.6 Opening a workspace

To open a workspace:

1. Select [Browse to another project workspace] option from the [Welcome!] dialog box and click the [OK] button or select [File->Open Workspace…]. The [Open Workspace] dialog box will be displayed.
2. Select the workspace file that you want to open.
3. Click the [Open] button to open the workspace. If the HEW is set up to display information when a workspace is opened, the [Workspace Properties] dialog box will be displayed. Otherwise, the workspace will be opened.

Note that whether the [Workspace Properties] dialog box is shown depends on the setting of either the [Show workspace information on workspace open] check box in the [Workspace Properties] dialog box or the [Display workspace information dialog on opening workspace] check box on the [Workspace] tab of the [Option] dialog box. Click the [OK] button in the [Workspace Properties] dialog box to open the workspace. Click the [Cancel] button to stop opening the workspace.

The HEW keeps track of the last five workspaces that you have opened and adds them to the [File] menu under the [Recent Workspaces] submenu. This gives you a shortcut to opening workspaces, which you have used recently.

To open a recently used workspace:

Select [Open a recent project workspace] in the [Welcome!] dialog box, select the name of the workspace from the drop-down list, and then click the [OK] button.

Another way is to select [File -> Recent Workspaces], and then from this submenu select the name of the workspace.

Note:

The HEW only permits one workspace to be open at a time. Consequently, if you attempt to open a second workspace, the first will be closed before the new one is opened.

1.7 Using old workspaces

The HEW can open any workspace that was created on a previous version of the HEW. This should be automatically upgraded when the workspace is opened. A back-up version of the initial workspace or project file must be saved in the current directory of the file that has been upgraded.

1.8 Saving a workspace

To save a HEW workspace, select the [File->Save Workspace] menu option.

1.9 Closing a workspace

To close a HEW workspace, select the [File->Close Workspace] menu option. If there are any outstanding changes to the workspace or any of its projects you will be requested whether or not you wish to save them.

1.10 Exiting the HEW

The HEW can be exited by selecting the [File–>Exit] menu option, pressing ALT+F4, or by selecting the close option from the system menu (which is opened by clicking the icon at the upper-left corner of the HEW title bar).

1.11 Component System Overview

The HEW allows the user to extend the HEW functionality by adding additional components to the system. This is achieved by registering the component in the [Tools Administration] dialog box. These components can add windows, menus and toolbars to the HEW system. Examples of the components are the debugger and builder components of HEW. The debugger component adds all of the menus and toolbars associated with the debugger and the builder component does the same for the build functionality. The components you have registered in the system will modify the look and feel of HEW. In some cases you may not have some of the menus which you can see in this manual. For instance if the builder component is not installed you will not have the toolchain menu option in the [Build] menu.

1.12 Debugger sessions

The HEW stores all of your builder options into a configuration. In a similar way, the HEW stores your debugger options in a session. The debugging platforms, the programs to be downloaded, and each debugging platform's options can be stored in a session.

Sessions are not directly related to a configuration. This means that multiple sessions can share the same download module and avoid unnecessary code rebuilds. Each session's data should be stored in a separate file in the HEW project.

See section 2.6, Build configurations, for more information about a configuration.

See section 15.1.5, Debugger sessions, for more information about a Debugger sessions.

2. Build basics

The HEW Builder is a Graphical User Interface designed to ease the development and building of applications written in C/C++ and assembly language for Renesas microcomputers.

Key Features:

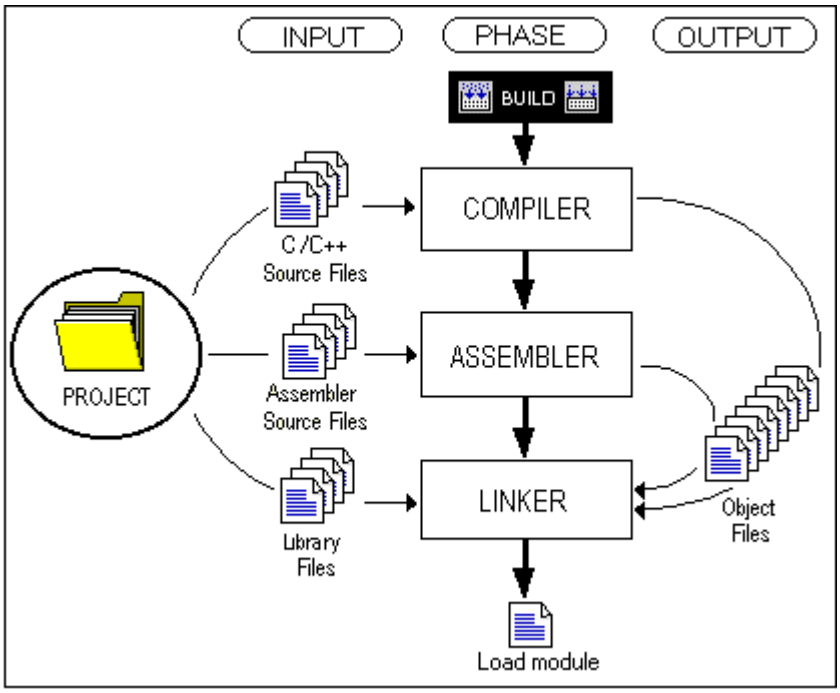
- Intuitive interface
- On-line help
- Common 'Look & Feel'

Notes:

- For detailed information about debugging platform hardware, please refer to the separate target debugging manual.

2.1 The build process

The typical build process is outlined in the figure below. This may not be the exact build process that your installation of HEW will use, as it depends upon the tools that were provided with your installation of HEW (you may not have a compiler, for instance). In any case, the principles are the same – each phase of the build takes a set of project files and builds them; if every file builds successfully then the next phase is executed.



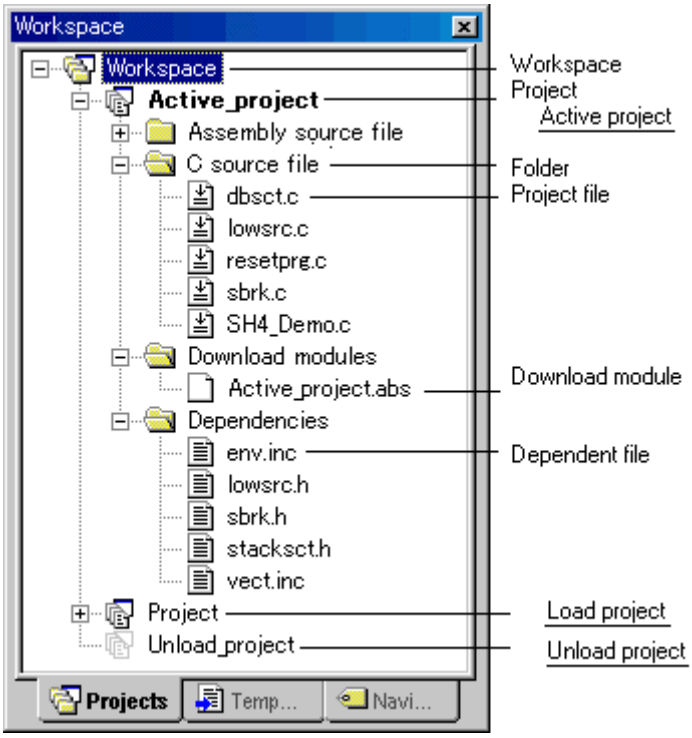
In the example shown in the figure above, the Compiler is the first phase, the Assembler is the second phase and the Linker is the third and final phase. During the Compiler phase, the C/C++ source files from the project are compiled in turn. During the Assembler phase, the assembler source files are assembled in turn. During the Linker phase, all library files and output files from the Compiler and Assembler phases are linked together to produce the load module.

The build process can be customized in several ways. For instance, you can add your own phases, disable phases, delete phases and so on.

These advanced build issues are left to chapter 3, Advanced build features.

2.2 Configuring the workspace window ([Projects] tab)

The **Projects** tab shows the current workspace, projects and files. You can quickly open any project file or dependent file by double-clicking on its icon.



Select a workspace on the [Projects] tab of the workspace window and click the right mouse button. This opens a pop-up menu containing the available options.

Pop-up Menu Options	Function
Insert Project...	Add project to workspace
Version Control	Executing version control tool
Configure View...	Configuring the workspace view
Properties	Display workspace properties

Select a project on the [Projects] tab of the workspace window and click the right mouse button. This opens a pop-up menu containing the available options.

Pop-up Menu Options		Function
Build	Build	Build out of date project files
	Build All	Build project files, regardless of whether the project files are out o date
	Update All Dependencies	Update all dependencies
Set as Current Project		Set as current project
Remove Project		Remove project from workspace
Unload Project OR Load Project		Unload the selected project OR Load the selected project
Add Files...		Add file(s) to a project
Remove Files...		Remove file(s) from project
Add Folder...		Add folder to a project
Version Control		Executing version control tool
Configure View...		Configuring the workspace view
Properties		Display project properties

Select a folder on the [Projects] tab of the workspace window and click the right mouse button. This opens a pop-up menu containing the available options.

Pop-up Menu Options	Function
Remove Folder	Remove folder
Rename Folder	Rename folder
Configure View...	Configuring the workspace view

Select a project file on the [Projects] tab of the workspace window and click the right mouse button. This opens a pop-up menu containing the available options.

Pop-up Menu Options	Function
Open	Open the selected file
Build	Build the selected file
Build Options	Set build options
Add File	Add file to a project
Remove File	Remove file from project
Exclude Build OR Include Build	Excluding a project file from build OR Including a project file in build
Version Control	Executing version control tool
Configure View...	Configuring the workspace view
Show Differences...	Compare files
Properties	Display file properties

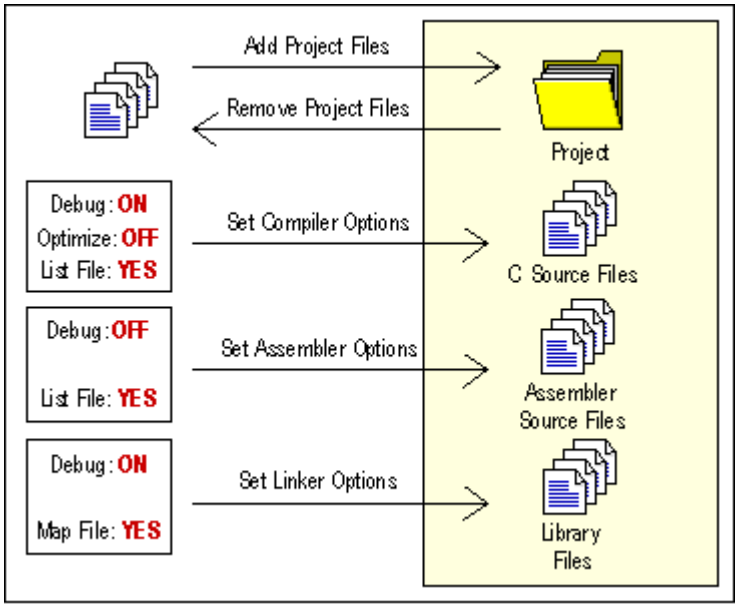
Select a download module on the [Projects] tab of the workspace window and click the right mouse button. This opens a pop-up menu containing the available options.

Pop-up Menu Options	Function
Download module	Download module
Download module (debug data only)	Download module (debug data only)
Unload module	Unload module
Configure View...	Configuring the workspace view

For details on Active project, Load project and Unload project, refer to section 2.11, Setting the current project.

2.3 Project files

In order for the HEW to be able to build your application, you must first tell it which files should be in the project and how each file should be built (see the figure below).



2.3.1 Adding files to a project

To add files to a project:

1. Select one of the following operations:
   - Select the [Project–>Add Files...] menu option, **OR**
   - Select the [Add Files...] menu option from the **Workspace** window's pop-up menu, **OR**
   - Press INS when the **Workspace** window is selected.
2. The [Add File(s)] dialog box will be displayed.
3. Select the file(s) that you want to add and click the **Add** button.

There are other ways to add files to a project:

- Right-clicking on an open file in the **Editor** window displays a pop-up menu. If the file is already in the project then the [Add File To Project] menu option is disabled. Select the [Add File To Project] menu option to add the file to the current project.
- In the HEW it is also possible to 'Drag and Drop' files from Windows® Explorer onto the **Workspace** window. These files will be automatically added to the project and displayed in the folder that they were dragged to.

Note:

If you add a file that has an unrecognized file type to the project, then the file will be added to the project, but certain functions will be disabled for this file. When you double-click on a file with an unrecognized file type in the **Workspace** window, the open operation is passed to the Windows® operating system (instead of opening the file in the editor). The default 'Open' operation is then carried out as if the file was double-clicked in Windows® Explorer. To view the currently defined extensions, use the [File Extensions] dialog box. See section 2.4, File extensions and file groups, for further information.

You can make the file a "Relative project file" if you click the [Relative Path]check box on the bottom of [Add File(s)] dialog box. This makes the project files relative and can be located outside of the workspace structure. Then if the entire source tree is relocated the HEW checks the relative location for the file and can still find the files. By default in this option is checked.

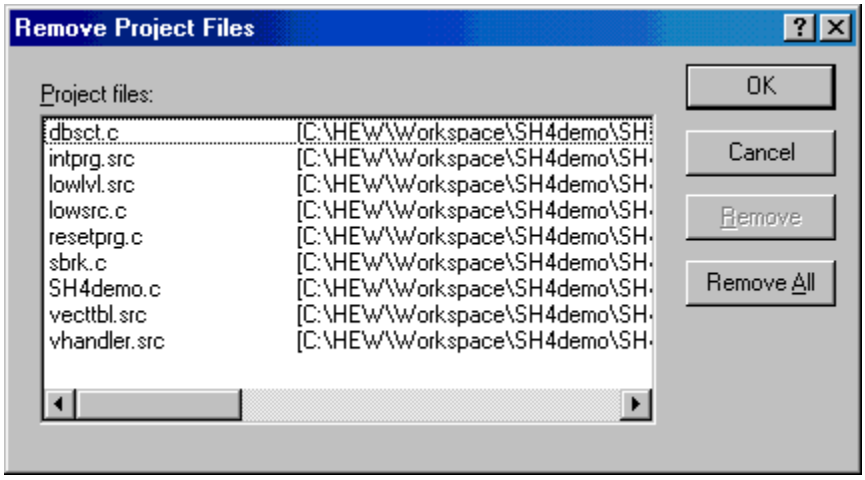
2.3.2 Removing files from a project

There are three ways of removing files from a project:

- Files can be individually removed from a project,
- A selection of files can be removed,
- All files can be removed.

To remove a file(s) from a project:

1. Select the [Project–>Remove Files...] menu option or select the [Remove Files...] menu option from the **Projects** tab's pop-up menu. The **Remove Project Files** dialog will be displayed.
2. Select the file(s) that you want to remove from the **Project files** list.
3. Click the **Remove** button to remove the file(s), or click the **Remove All** button to remove all files from the list.
4. Click the **OK** button to remove the files from the project.



To remove selected files from a project using the Workspace window:

1. Select the files that you want to remove in the **Projects** tab of the **Workspace** window. Multiple files can be selected by holding down the SHIFT or CTRL key.
2. Press DEL. The selected files will be removed.

2.3.3 Excluding a project file from build

A file in a project can be individually excluded from build on a configuration by configuration basis.

To exclude a project file from build:

1. Right-click on the file that you want excluded from build, in the **Projects** tab of the **Workspace** window.
2. Select the [Exclude Build <file>] option (where <file> is the name of the selected file) from the pop-up menu. A red cross  will appear on the file's icon, and the file will be excluded from build.

2.3.4 Including a project file in build

A file that has been excluded from build can be included again.

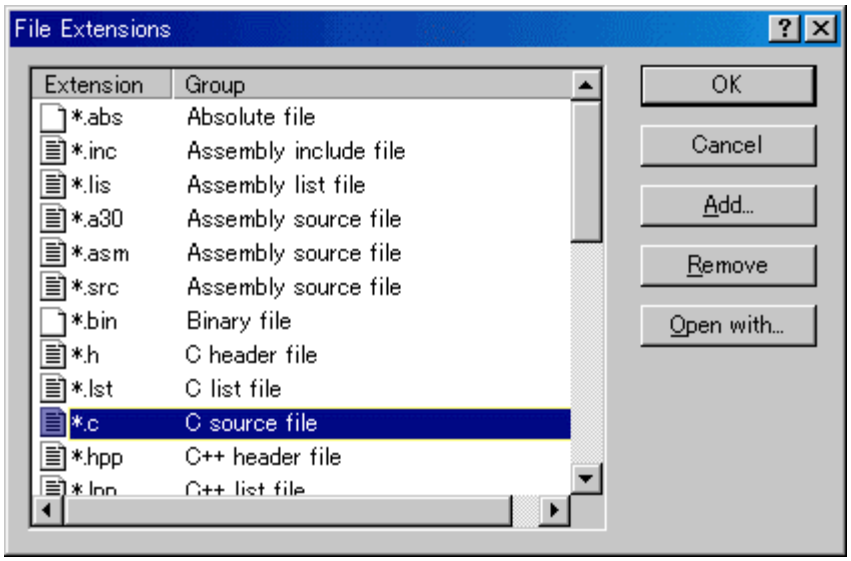
To include a project file in build:

1. Right-click on a file that has previously been excluded from build, in the **Projects** tab of the **Workspace** window.
2. Select the [Include Build <file>] option (where <file> is the selected file) from the pop-up menu. The red cross will be removed from the file's icon  , and the file will be included in build.

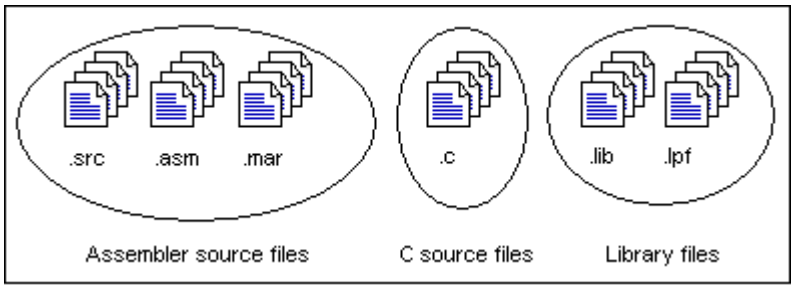
2.4 File extensions and file groups

The HEW can identify files by their extension. The system defines certain extensions depending upon the tools that are being used. For example, if you are using a compiler then the `.c` extension will be in the 'C source file' group and will be used as input to the compiler phase. Additionally, the HEW allows you to define your own extensions. For example, if the project you are developing uses assembler source files the default extension may be `.src`. If you would like to use a different extension instead of `.src` (e.g. `.asm`) then you can define a new extension and request that the HEW treats it in the same way as a `.src` file.

File extensions and file groups can be viewed and modified via the **File Extensions** dialog, which is invoked by selecting the [Project–>File Extensions] menu option. This dialog displays all the extensions and file groups that are defined within the current workspace.

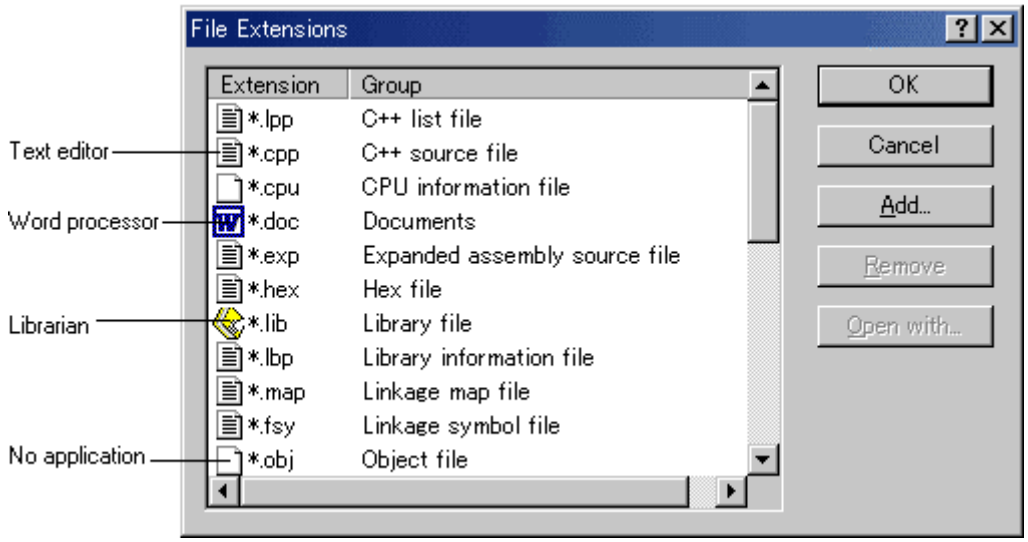


The **File Extensions** list is divided into two columns. On the left are the file extensions, and on the right are the file groups associated with the extensions. Many file extensions can belong to the same group. For example, assembler source files may have several extensions in a single project (e.g. `.src`, `.asm`, `.mar` etc.).



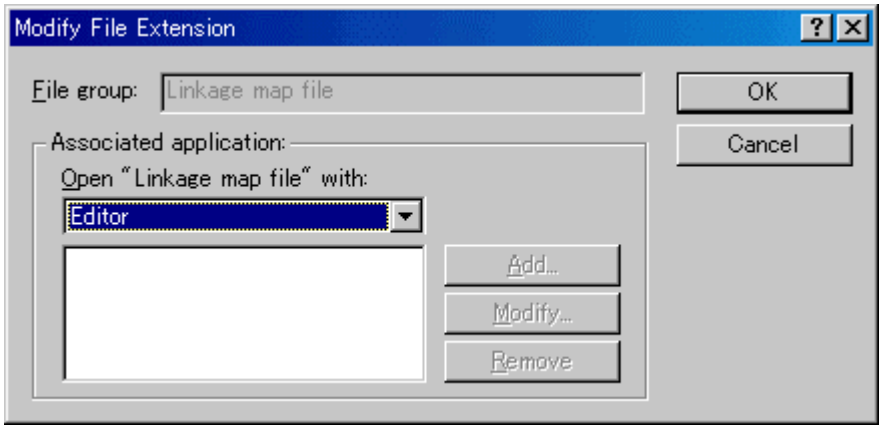
2.4.1 Associating an application with a file group

In addition to opening a file with the editor, the [File Extensions] dialog allows you to associate any application with any file group so that when you double-click on a file in the **Projects** tab of the **Workspace** window, the appropriate application is launched with the file.

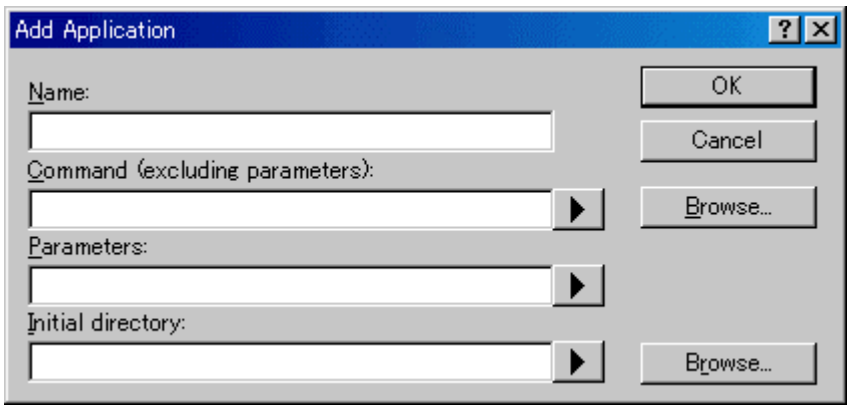


To associate an application with a file group:

1. Select the [Project–>File Extensions...] menu option from the menu bar. The [File Extensions] dialog will be displayed.
2. Select the file group that you want to associate from the file extensions list.
3. Click the [Open with...] button. The [Modify File Extension] dialog will be displayed.



4. Select **None** to remove any association, **Editor** to open this type of file in the internal/external editor, or select **Other** to open this type of file with a specific application. If you select **Other**, you can either specify a new application, or select any previously defined application from the drop-down list. Click the **Add** button to define a new application. The [Add Application] dialog will be displayed.



Enter the name of the tool into the **Name** field. Enter the full path to the tool in the **Command** field (do not include any parameters). Enter the parameters that are required to open a file into the **Parameters** field. Be sure to use the `$(FULLFILE)` placeholder to specify the location of the file (see Reference 4, Placeholders, for more information on placeholders). Enter the initial directory in which you would like the application to run into the **Initial directory** field. Click the **OK** button to finish creating the application. Click the **Modify** button to modify an application. The [Modify Application] dialog will be displayed. This dialog is the same as the [Add Application] dialog described above except that the **Name** field is read-only. Modify the settings as desired and then click the **OK** button.

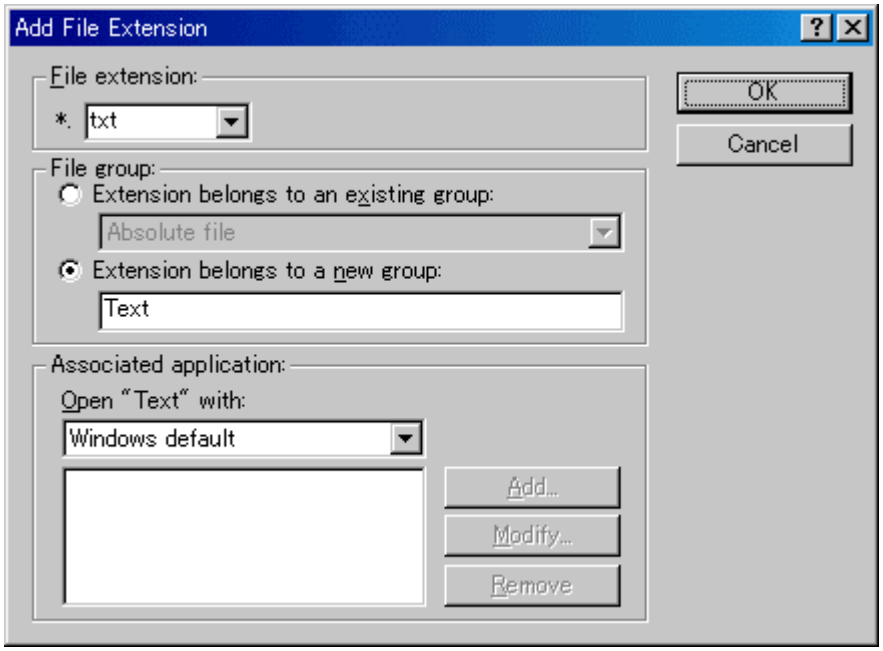
5. Click the **OK** button to set the application for the selected file group.

2.4.2 Creating a new file extension and file group

If you want to manage files that are not, by default, recognized by the HEW (e.g. documents) then you need to create a new extension and a new file group.

To create a new file extension in a new file group:

1. Select the [Project–>File Extensions...] menu option from the menu bar. The [File Extensions] dialog will be displayed.
2. Click the **Add** button. The [Add File Extension] dialog will be displayed.



3. Enter the extension that you want to define into the **File extension** field. Use only alphanumerics and an underscore as characters of a file extension string. The drop-down list contains all extensions that are undefined in the current project. Selecting one of these extensions will add the text to the file extension field automatically.
4. Select the **Extension belongs to new group** option and enter a description that defines this new file group.
5. At this stage it is possible to change the associated application. There are four available choices in the **Open <extension group> with** drop-down list:
   - Editor
   - None
   - Other
   - Windows default

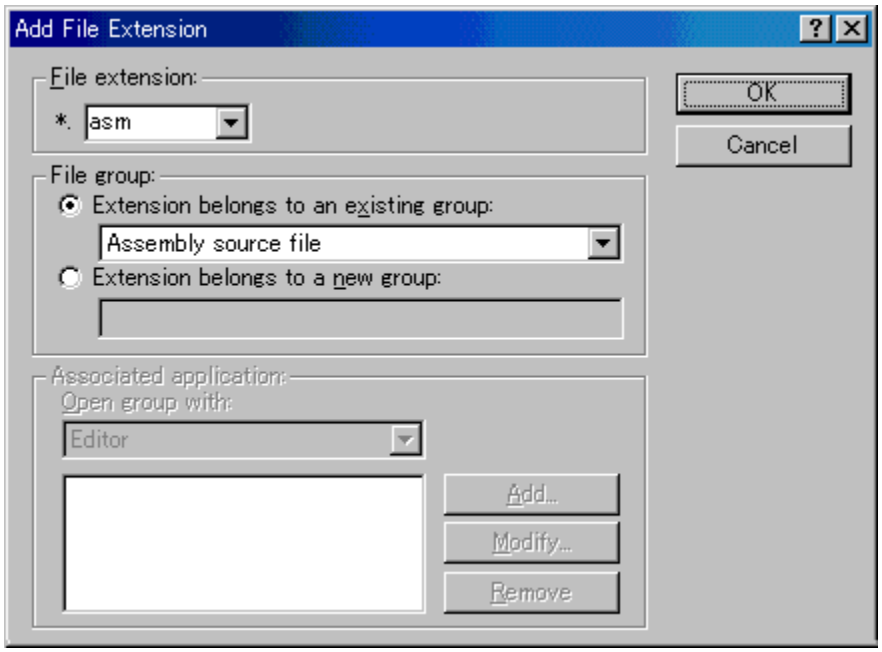
   If **Editor** is selected, the **Open File** function in the workspace window causes the file to be opened in the HEW editor. If **None** is selected then the **Open File** operation is disabled when it is attempted. Selecting **Other** allows you to configure another tool for the **Open File** operation. See section 2.4.1, Associating an application with a file group, for more details. If **Windows default** is selected then the **Open File** function in the **Workspace** window passes the **Open File** operation to the Windows® operating system. This then selects the default behavior for this file extension as defined in Windows® Explorer.
6. Click the **OK** button to add the extension to the **File Extensions** list.

2.4.3 Creating a new file extension

If your files use a different extension from those accepted by the HEW for a given phase (e.g. your assembler source files are `.asm` but the HEW only recognizes `.src`), then you need to create a new extension and add it to an existing file group. This process is described below.

To create a new file extension in an existing file group:

1. Select the [Project–>File Extensions...] menu option from the menu bar. The [File Extensions] dialog will be displayed.
2. Click the **Add** button. The [Add File Extension] dialog will be displayed.



3. Enter the extension that you want to define into the **File extension** field. Use only alphanumerics and an underscore as characters of a file extension string. The drop-down list contains all extensions that are undefined in the current project. Selecting one of these extensions will add the text to the file extension field automatically.
4. Select the **Extension belongs to an existing group** option and select the group to which you would like to add this new extension.
5. Click the **OK** button to add the extension to the **File Extensions** list.

2.5 Setting build options

Once you have added the necessary files to the project, the next step is to instruct the HEW on how to build each file. To do this, you will need to select a menu option from the [Build] menu. The contents of this menu depend upon which tools you are using.

To set options for a build phase:

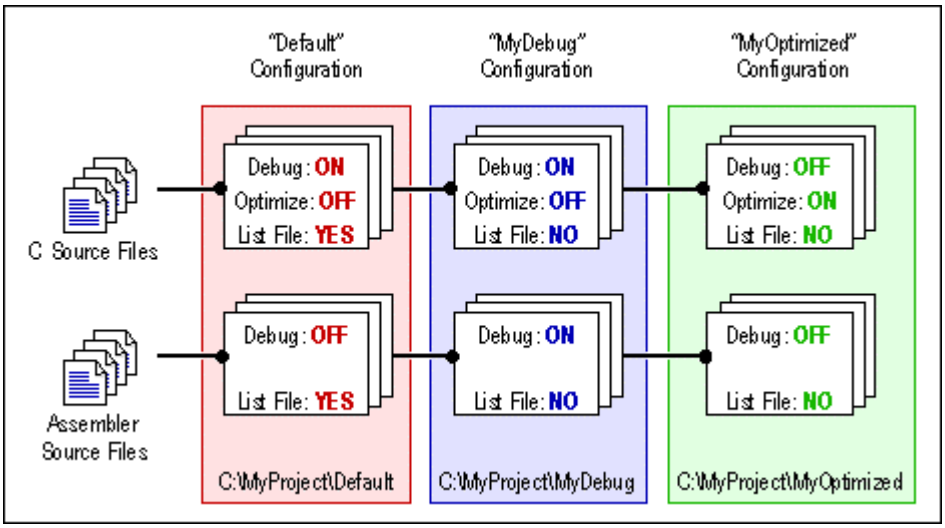
1. Select the [Build] menu and select the phase whose options you would like to modify.
2. A dialog will be displayed allowing you to specify the options.
3. After making your selections, click the [OK] button to set them.

To obtain further information, use the context-sensitive help button or select the area in which you need assistance and press F1.

2.6 Build configurations

The HEW allows you to store all of your build options into a build configuration, i.e. you can 'freeze' all of the options and give them a name. Later on, if you select that configuration, all options for all of the build phases will be restored. These configurations also allow the user to specify debugger settings for a build configuration. This means that each configuration can be targeted at a different end platform.

The figure below shows three configurations: **Default**, **MyDebug** and **MyOptimized**. In the first configuration, **Default**, each phase (compile and assemble) is set to its standard settings. In the second configuration, **MyDebug**, each file is being built with debug information switched on. In the third configuration, **MyOptimized**, each file is being built with optimization on full and without any debug information. The developer of this project can select any of those configurations and build them without having to return to the options dialogs to set them again.



2.6.1 Selecting a build configuration

To select the current configuration:

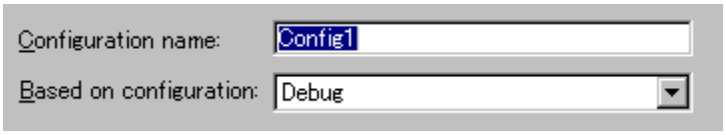
1. Select the [Build–>Build Configurations...] menu option to display the [Build Configurations] dialog.
2. Select the build configuration that you want to use from the **Current Configuration** drop-down list.
3. Click the **OK** button to select the configuration.

You can also select a different build configuration by selecting it from the **Current Configuration** drop-down list box on the [Standard] toolbar.

2.6.2 Adding a new build configuration

To add a new build configuration:

1. Select the [Build–>Build Configurations...] menu option to display the [Build Configurations] dialog.
2. Click the [Add...] button. The [Add Configuration] dialog will be invoked.
3. Enter the new build configuration name into the **Configuration name** field. As you enter the new build configuration name, the directory underneath changes to reflect the configuration directory that will be used. Select one of the existing build configurations, on which you want to base the new build configuration, from the **Based on configuration** drop-down list. Click the **OK** buttons on both dialogs to complete the creation of the new build configuration.



2.6.3 Removing a build configuration

To remove a build configuration:

1. Select the [Build–>Build Configurations...] menu option to display the [Build Configurations] dialog.
2. Select the build configuration to remove and click the **Remove** button.
3. Click the **OK** button to close the [Build Configurations] dialog.

2.7 Building a project

2.7.1 Building individual files

The High-performance Embedded Workshop lets you build project files individually.

To build an individual file:

1. Select the file to build from the **Projects** tab of the **Workspace** window.
2. Select one of the following operations:
   - Click the [Build File] toolbar button  , **OR**
   - Select the [Build <file>] option from pop-up menu, **OR**
   - Select the [Build–>Build File] menu option, **OR**
   - Press CTRL+F7.

All output is redirected to the **Build** tab of the Output window.

2.7.2 Building a project

The [Build] option only compiles or assembles those files that have changed since the last build. Additionally, it will rebuild source files if they depend upon a file that has changed since the last build. For instance, if the file '`TEST.C`' `#include`'s the file '`HEADER.H`' and the latter has changed since the last build, the file '`TEST.C`' will be recompiled.

To perform a build operation:

Select one of the following operations:

- Click the Build toolbar button  , **OR**
- Press F7, **OR**
- Select the [Build–>Build] menu option, **OR**
- Right-click on a project in the **Projects** tab of the **Workspace** window and select the [Build->Build] option from the pop-up menu.

The [Build All] option compiles and assembles all source files, irrespective of whether they have been modified or not, and links all of the new object files produced.

To perform a build all operation:

Select one of the following operations:

- Click the Build All toolbar button  , **OR**
- Select the [Build–>Build All] menu option, **OR**
- Right-click on a project in the **Projects** tab of the **Workspace** window and select the [Build->Build All] option from the pop-up menu.

All output from a build or build all operation is redirected to the **Build** tab of the Output window. Both the Build and the Build All operations will terminate if any project files produce errors.

2.7.3 Stopping a build

The High-performance Embedded Workshop allows you to halt the build process once it is under way.

To stop a build:

1. Select one of the following operations:
   - Click the [Stop Build] toolbar button  , **OR**
   - Select the [Build–>Stop Build] menu option.

   The build will stop after the current file/phase has been built.
2. Wait until the 'Build Stopped by User' message appears in the **Output** window before continuing.

To forcibly terminate a task:

Select the [Build→Terminate Current Tool] menu option. The HEW will attempt to stop the tool immediately.

Note:

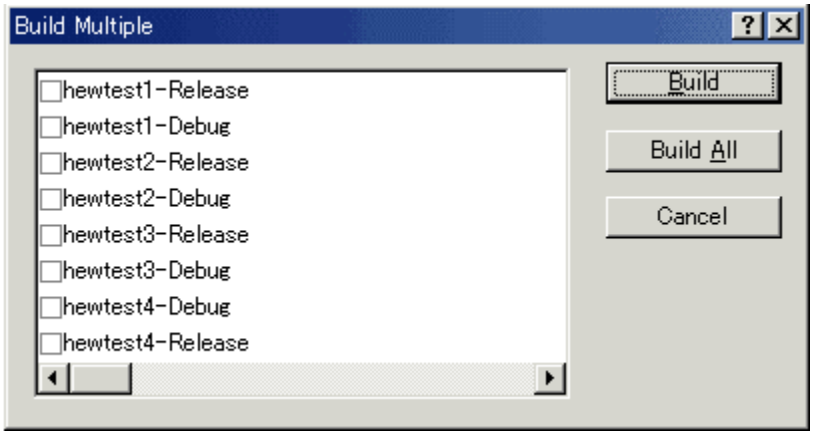
Do NOT assume that any output from the tool you terminated is valid. It is recommended that you delete any output files produced and ensure that the phase is executed again.

2.7.4 Building multiple projects

The High-performance Embedded Workshop lets you build multiple projects and configurations at once.

To build multiple projects:

1. Select the [Build–>Build Multiple...] menu option.
2. The [Build Multiple] dialog box gives you the choice of which projects and configurations to build. Select the check boxes next to the projects and configurations that you want to build. For example, in figure below if you wanted to build the entire "hewtest2" project you would check the "hewtest2-Debug" and the "hewtest2-Release" selections and leave all other check boxes unchecked.



3. When you are happy with your chosen selection, click the **Build** button and the HEW will build the selected projects and configurations.

4. If you want to build all of the projects, click the **Build All** button. This will automatically select all projects and configurations, and build them all.
5. Results from the build are displayed in the **Build** tab of the Output window in the same way as in a normal build process.

2.7.5 The Output Window

When a tool executes (i.e. compiler, assembler, linker etc.) its output is displayed in the “Output” window. If any of the tools produce any errors or warnings then they are displayed along with the source file name and the line number at which the error is located. To quickly locate a specific bug, double click on a given error/warning to invoke the current editor.

If you right-click anywhere inside the output window, a pop-up menu will be invoked.

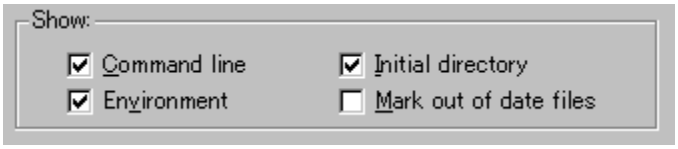
Pop-up Menu Options	Function
Display next Error/Warning	Display next error or warning.
Display previous Error/Warning	Display previous error or warning.
Help	Display help information for line.
Go to Error/Warning	Go to the associated source line.
Clear Window	Clear the window.
Save	Save the content of the window.

2.7.6 Controlling the content of the output window

It is often useful to display extra information (such as the command line options that are being applied to a file) during a build. The HEW allows you to specify whether or not you want such options displayed in the **Build** tab of the Output window during a Build, Build All or Build File operation via the [Options] dialog.

To view or hide extra information during a build:

1. Select the [Setup–>Options] menu option. The **Setup Options** dialog will be displayed. Select the **Build** tab.
2. Set the three check boxes in the **Show** group as follows:
   - **Command line** controls whether the command line is shown as each tool is executed.
   - **Environment** controls whether the environment is shown as each tool is executed.
   - **Initial directory** controls whether the current directory is shown as each tool is executed.

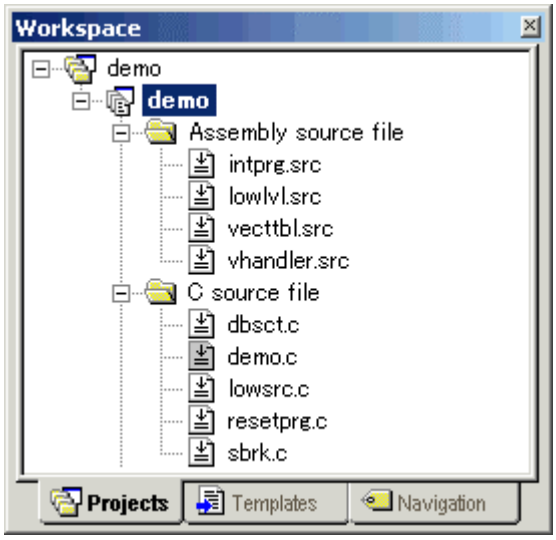


2.7.7 Displaying out of date files in the workspace window

Files updated later than the file generated by the previous build (i.e. out of date files) are marked in the workspace window. In the window below the file "demo.c" is out of date.

When you click [Build] next time these files will be re-built. This is also displayed for dependent projects of the current project.

The view of these files is updated whenever something that affects the build occurs, e.g. options changing, file addition, dependencies changing, files modified, etc.



To display out of date files in the workspace window:

1. Click on the [Setup->Options...] menu item.
2. Select the [Build] tab.
3. Check the [Mark out of date files] check box.
4. Click [OK].

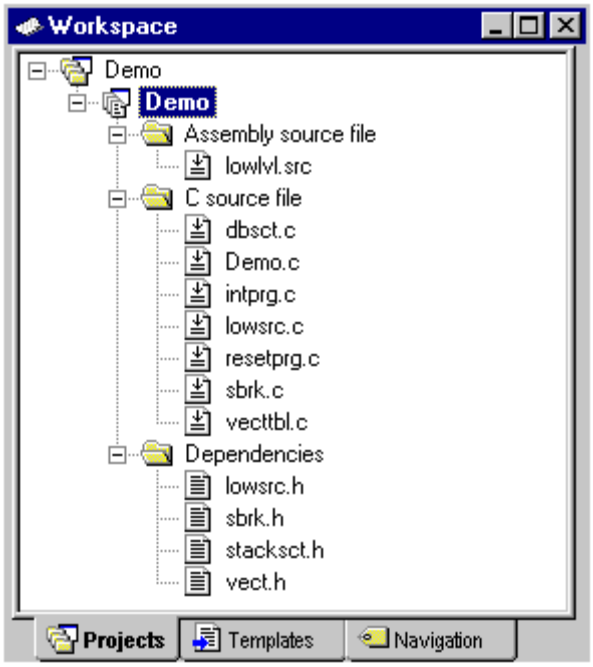
2.8 File dependencies

A typical project will contain dependencies between files. For example, a C file may '`#include`' one or more header files. In complex projects, source files will include (or depend upon) others and this can quickly become difficult to manage. However, the HEW provides a dependency scanning mechanism whereby all files in a project are checked for dependencies. Once complete, the **Projects** tab of the Workspace window will display an up-to-date list with all the project file dependencies.

To update a project's dependencies:

- Select the [Build–>Update All Dependencies] menu option, **OR**
- Right-click on a project in the **Projects** tab of the **Workspace** window and select the [Build–>Update All Dependencies] option from the pop-up menu.

Initially, the dependencies for all files are contained within the **Dependencies** folder (although this can be modified by configuring the projects tab).



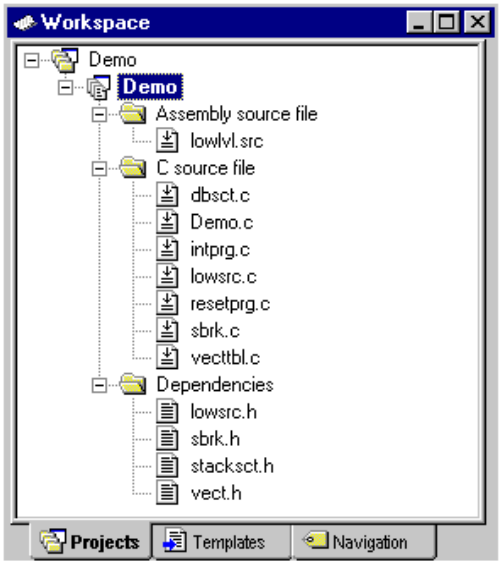
2.9 Configuring the workspace window

If you right-click anywhere inside the **Projects** tab of the **Workspace** window, a pop-up menu will be invoked. Select the [Configure View...] menu option to modify the way in which information is displayed. The following four sections detail the effect of each option on the [Configure View] dialog.

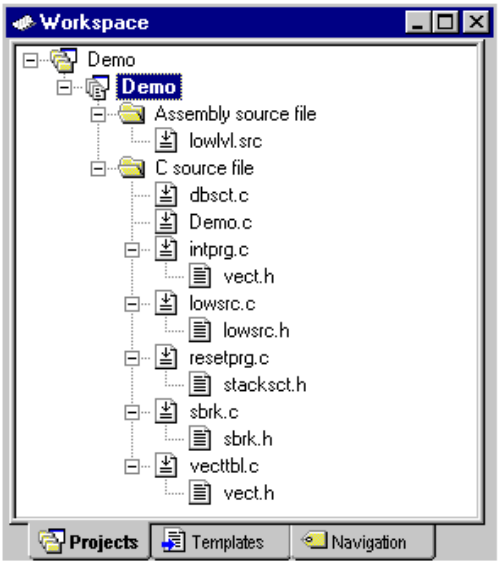
Show Dependencies Under Each File

If you check the [Show dependencies under each file] checkbox, the dependent files are shown under the including source file as a flat structure, i.e. the files themselves become folders (as in figure (ii) below). If this option is not selected then a separate folder contains all dependencies (as in figure (i) below).

(i)



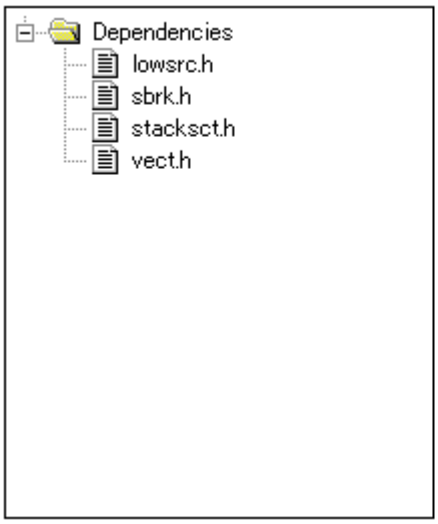
(ii)



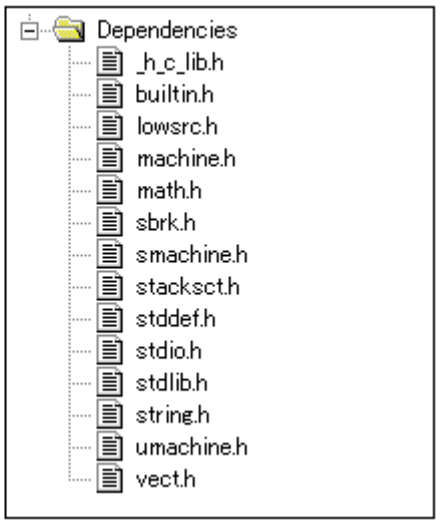
Show Standard Library Includes

By default, any dependent files found in standard include paths will not be shown (as in figure (j) below). For example, in C code, if you write a '`#include`' statement, such as '`#include <stdio.h>`', `stdio.h` will not be listed as a dependent file. To view such system include files, select the [Show standard library includes] checkbox (as in figure (jj) below).

(j)



(jj)

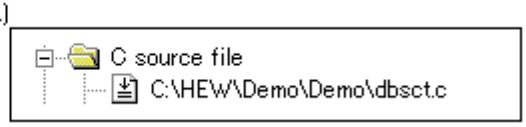


Show File Paths

By default, any files will not be shown with their full path (as in figure (k) below). If the [Show file paths] checkbox is selected, all of the files in the **Projects** tab of the **Workspace** window are shown with their full path, i.e. from a drive letter (as in figure (kk) below).

(k)

(kk)



Sorting the workspace window into time stamp order

When [Sort files in time order] option is selected, the workspace window is then sorted into file time stamp order. The newest files are located at the top of list with the older files towards the bottom.

If files are updated after selecting this option, manually update the order of these files.

To manually update the file order:

Select [Refresh Order] in the pop-up menu on the **Projects** tab of the **Workspace** window.

2.10 Inserting a project into the workspace

When a workspace is created, it contains only one project but, after it is created, you can insert new or existing projects into the workspace.

To insert a new project into the workspace:

1. Select the [Project–>Insert Project...] menu option. The [Insert Project] dialog box will be displayed.
2. Select the [New project] radio button.
3. Click the [OK] button. The [Insert New Project] dialog box will be invoked.
4. Enter the name of the new workspace into the [Project Name] field. This can be up to 32 characters in length and contain letters, numbers and the underscore character. Especially, do not use a minus sign, or a space. As you enter the project name the HEW will add a sub-directory for you automatically. This can be deleted if desired.
5. Use the [Browse...] button to graphically select the directory in which you would like to create the project. Alternatively, you can type the directory into the [Directory] field manually.
6. The project type list displays all of the available project types (e.g. Application, Library etc.). Select the type of project that you want to create from this list.
7. Click the [OK] button to create the project and insert it into the workspace.

To insert an existing project into a workspace:

1. Select the [Project–>Insert Project...] menu option. The [Insert Project] dialog box will be displayed.
2. Select the [Existing project] radio button.
3. Enter the full path of the project database file (`.HWP` file) into the edit field, or click the [Browse...] button to search for it graphically.
4. Click the [OK] button to insert the existing project into the workspace.

2.11 Setting the current project

A project can be in three states – the **Current** project, a **Loaded** project or an **Unloaded** project.

Since a workspace can contain many projects, only one of them can be the **Current** project at any time. This project is the one that build actions and debug operations can be performed on (e.g. clicking the Build toolbar button will build the **Current** project).

To set a project as the current project:

Select one of the following operations:

- Select the project that you want to make active from the [Project→Set Current Project] sub-menu, **OR**
- Select the project from the **Projects** tab of the **Workspace** window. Right-click to display the pop-up menu and select the **Set as Current Project** option.

Note:

It is also possible to select which project you want to make Current from the [Project→Set Current Project] sub-menu.

If the project is **Loaded**, it is possible to open the project's directory and view the files. It is also possible to change the builder or debugger options for the project. A **Loaded** project can also have tool executions performed on it from the [Tools] menu.

To load a project in the workspace:

1. Select the **Unloaded** project from the **Projects** tab of the **Workspace** window.
2. Right-click to display the pop-up menu and select the **Load Project** option.

If the project is **Unloaded**, its icon appears 'grayed' in the **Projects** tab of the **Workspace** window and no actions can be performed upon it.

To unload a project from the workspace:

1. Select the **Loaded** project from the **Projects** tab of the **Workspace** window.
2. Right-click to display the pop-up menu and select the **Unload Project** option.

Note:

It is possible to select multiple projects in the workspace window to perform these operations.

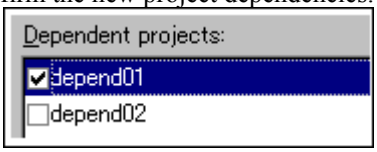
2.12 Specifying dependencies between projects

The projects within a workspace can be dependent upon one another so that when one project is built, all its dependent projects are built first. This is useful if one project uses another in the workspace. For example, imagine that a workspace contains two projects. The first project is a library, which is included by an application project. In this case the library must have been built and up-to-date before the second application can build correctly. To achieve this situation we can specify the library as a dependent (i.e. child) project of the application project. This would then allow the library to be built first if it is out-of-date.

When a dependent project is built, the HEW attempts to match the configuration in the dependent project with that of the current project. This means that if the current configuration is 'Debug' then the HEW will attempt to build the 'Debug' configuration in the dependent project. If this matched configuration does not exist then the HEW will use the configuration that was last used in the dependent project.

To make projects depend upon another:

1. Select the [Project→Dependent Projects...] menu option. The **Dependent Projects** dialog will be displayed.
2. Select the project to which you would like to add dependents. When you do this, the **Dependent Projects** list will display all of the projects in the workspace (excluding the selected project).
3. The **Dependent Projects** list has a checkbox for each project listed. Set the associated checkboxes to make those projects depend upon the selected project.
4. Click the **OK** button to confirm the new project dependencies.



2.13 Removing a project from the workspace

To remove a project from a workspace:

1. Select the project from the **Projects** tab of the **Workspace** window.
2. Select one of the following operations:
   - Press DEL, **OR**
   - Right-click on the selected project to invoke a pop-up menu. Select the [Remove Project] menu option.

Note:

You cannot remove the Current project from the workspace.

2.14 Relative projects paths in the workspace

In the High-performance Embedded Workshop when you add a project you can choose to add the project to the workspace using a relative path. This allows you to position a file above the workspace directory and it will still be relocated correctly if you relocate the HEW workspace. The project is always relative to the workspace so if the project is one directory above the workspace before it is moved the HEW will try to find the project in the same relative location after the relocation procedure. This is especially useful if you are using a project shared between more than one workspace.

In older versions of HEW this project would not have been relocated and would have still tried to access the original file path. The older version of HEW could only relocate the projects, which were in a sub-directory of the workspace directory. This is still the standard behavior for the High-performance Embedded Workshop.

To add the project to the workspace using a relative path:

1. Select the project in the workspace window.
2. Right click and then select **Properties**.
3. Click the [Project relative file path] checkbox to switch the relative file path feature.
4. Click OK.

2.15 User folders in the workspace

In the High-performance Embedded Workshop it is possible to add folders to your workspace window. This allows you to logically group your files into certain areas within a project. The folder can be set to any name and this is entered in a dialog.

To add a user folder:

1. Select the project on the “Projects” tab of the workspace window.
2. Right click and then add folder.
3. Enter the name and click OK.
4. You can now drag and drop files into this folder to group them logically.

To remove a user folder:

1. Select the folder on the “Projects” tab of the workspace window.
2. Right click and then select Remove folder. Note that the folder must be empty and that the delete key can also be used instead of the pop-up.

To modify a user folder name:

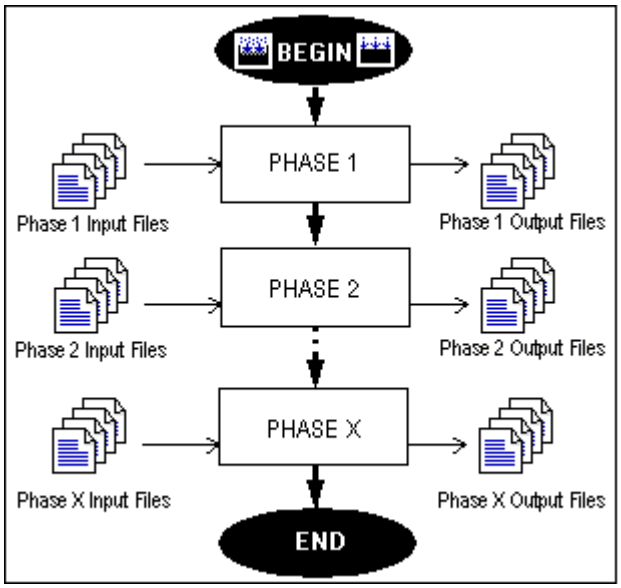
1. Select the folder on the “Projects” tab of the workspace window.
2. Right click and then select Modify folder name.
3. Enter the new name in the dialog.
4. Click “OK”.

3. Advanced build features

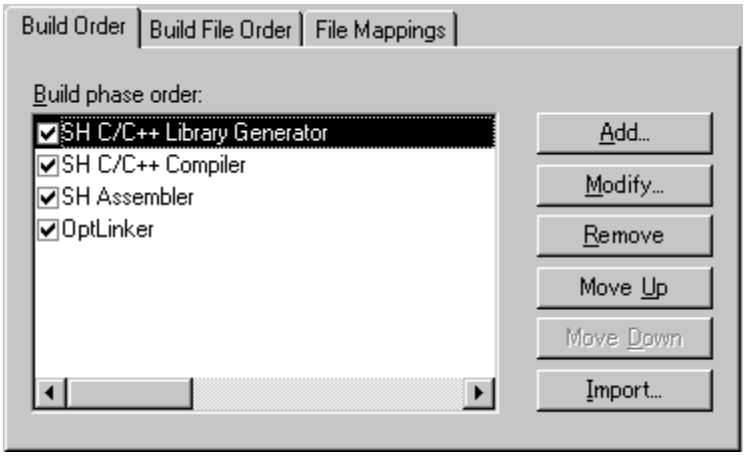
3.1 The build process revisited

3.1.1 What is a build?

Building a project means applying a set of tools upon certain input files in order to produce the desired output. Thus, we apply a compiler upon C/C++ source files in order to create object files, we apply an assembler upon assembler source files in order to create object files and so forth. At each step or 'phase' of the build, we apply a different tool upon a different set of input files. The figure below presents a different view of the build process.



The High-performance Embedded Workshop provides the ability to change this build process via its **Build Phases** dialog, which can be accessed by selecting the [Build–>Build Phases...] menu option. On the left-hand side are the phases that are defined in the current project.



Through this dialog you can create your own phases, modify phases, remove phases, enable/disable phases, change the order of phases and so on.

3.2 Creating a custom build phase

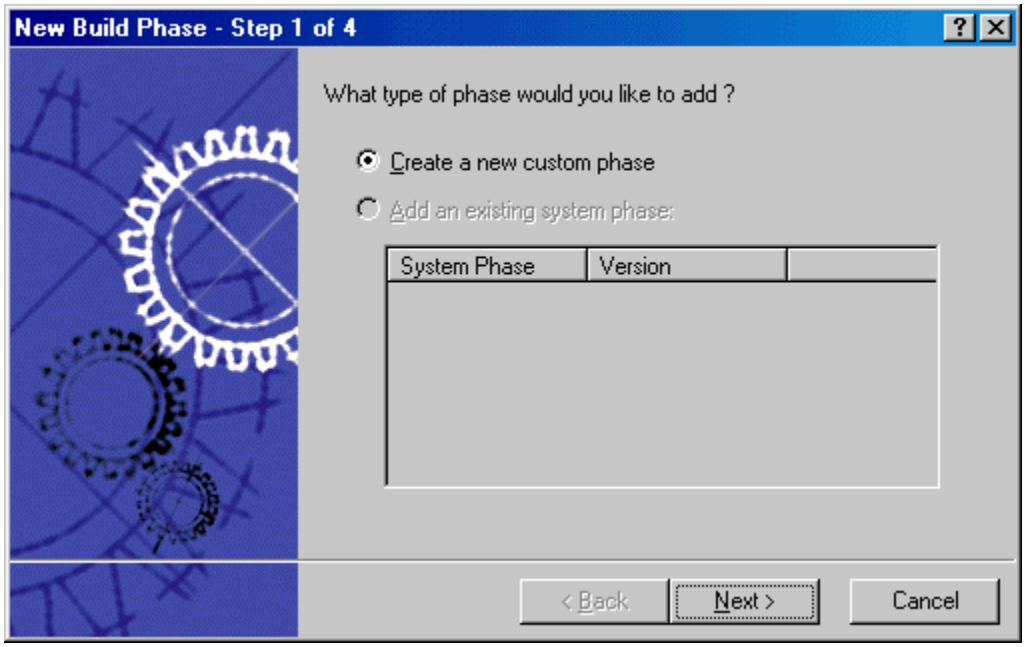
If you want to execute another tool before, during or after a standard build process then this can be achieved by creating your own (i.e. custom) build phase.

To create a new custom build phase:

1. Select the [Build–>Build Phases...] menu option to invoke the **Build Phases** dialog.
2. Click the [Add...] button. This will invoke the **New Build Phase** wizard dialog.
3. Follow the 4 steps below. To move forward and backward between steps click the [Next >] and [< Previous] buttons respectively.

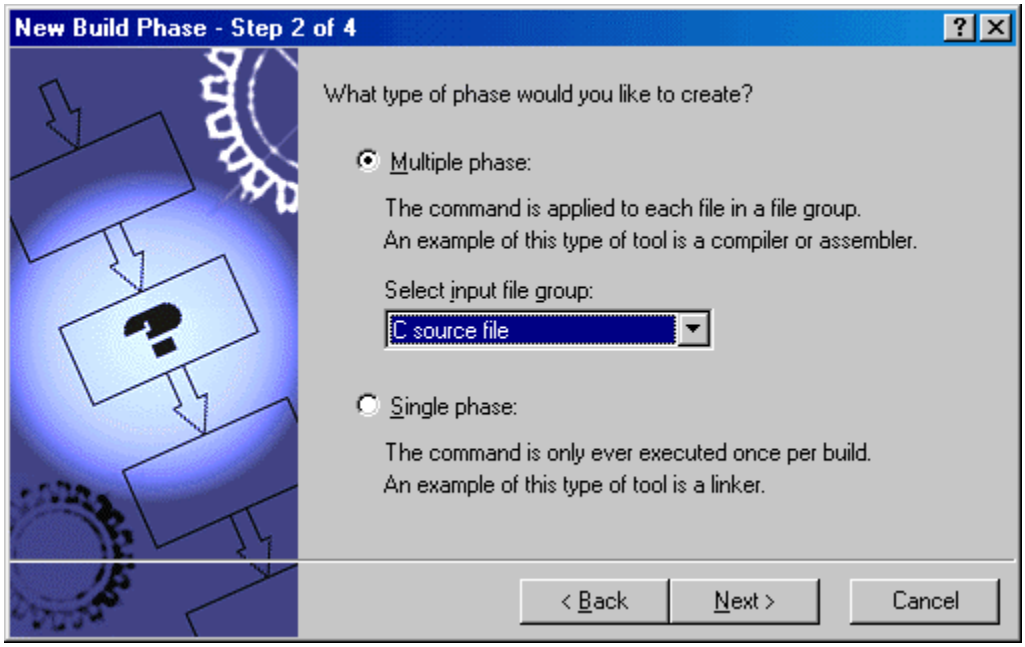
Step 1

The first step asks whether you want to create an entirely new phase or whether you want to add a system phase. A system phase is a 'ready-made' phase which is already defined within the toolchain you are using (e.g. compiler, assembler, linker, librarian, etc.) or a utility phase (e.g. file copy, complexity analyzer etc.). The **Add an existing system phase** button is inactive if no more system phases are available. Select the **Create a new custom phase** button to create your own build phase.

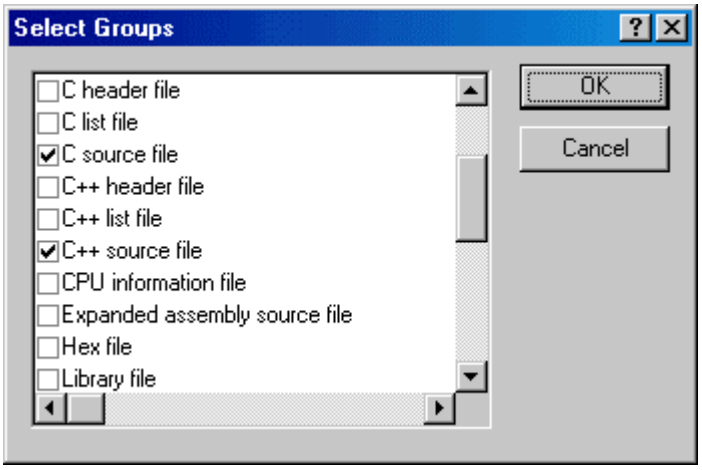


Step 2

The second step asks what type of phase you would like to create. There are two choices: multiple or single. When a multiple phase is executed, the command is applied to each file in the project of a certain file group. For example, if you set the input file group to be C source files then the command will be executed once for each C source file in the project. A single phase is executed once at most during a build.



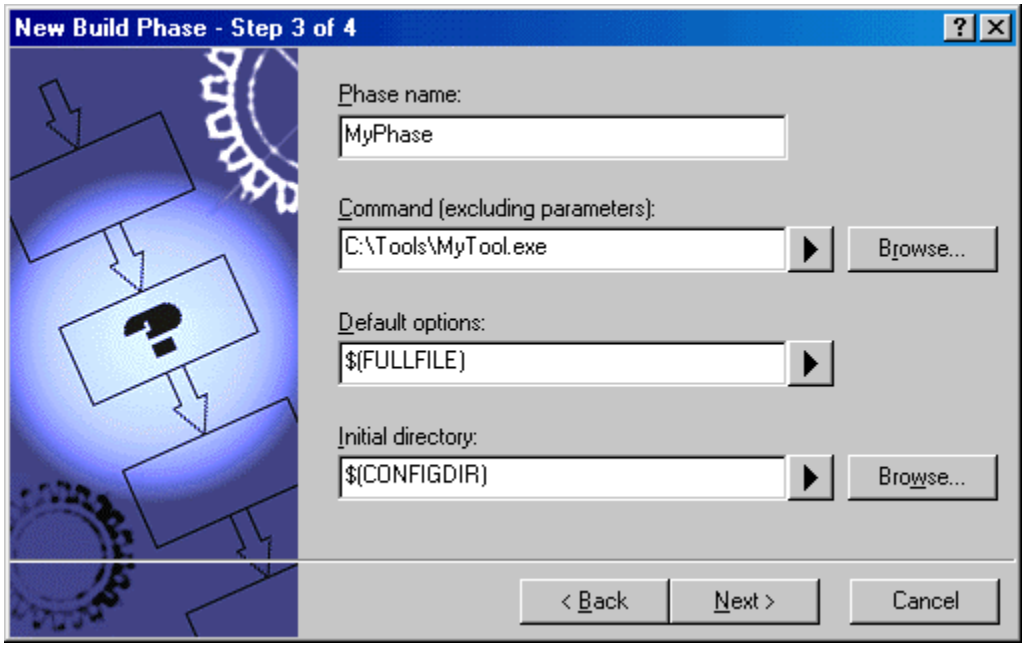
The input file group list contains the current file groups defined for the project. It is possible to define multiple input file groups by selecting the **Multiple Groups** entry in the input file group list.



Once this choice has been made the input file group selection is displayed as **Multiple Groups**. This dialog allows the user to choose multiple input file groups for the custom phase being added to the project. To select a file group check the box next to the file group's name. One or more file groups can be selected in this dialog.

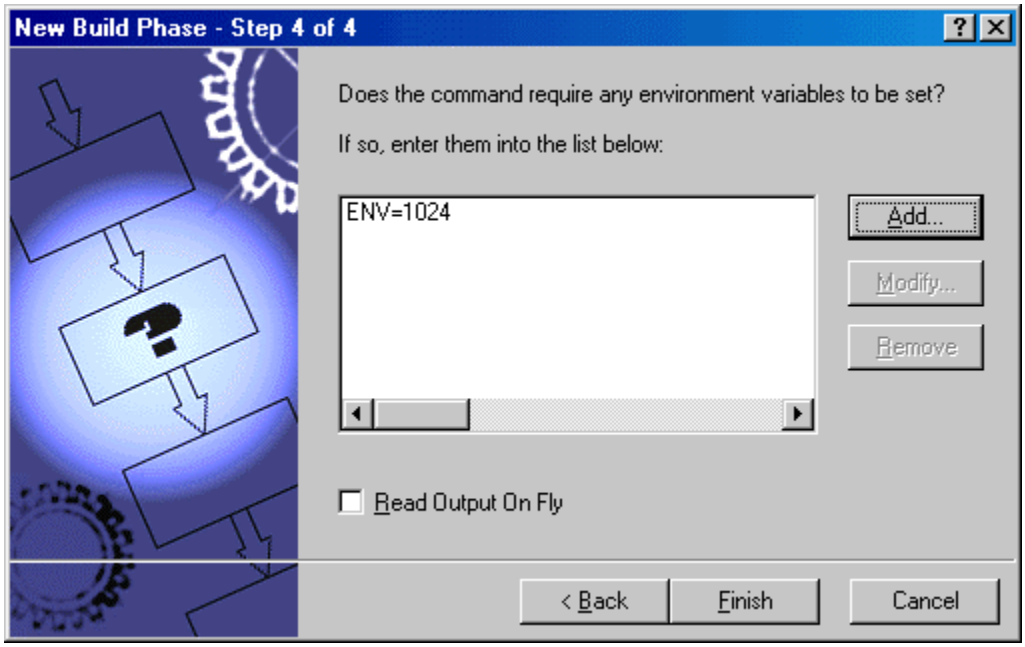
Step 3

The third step requests the fundamental information about the new build phase. Enter the name of the phase into the **Phase name** field. Enter the location of the program file into the **Command** field (do not insert any command line options as these options are specified via the [Options] menu of the HEW menu bar). Specify the default options for the phase (i.e. what options you would like new files to take when added to the project) into the **Default Options** field. If you have a preferred directory in which you would like this program to run (i.e. where you want the current working directory to be set to before the tool is executed) then enter it into the **Initial directory** field.

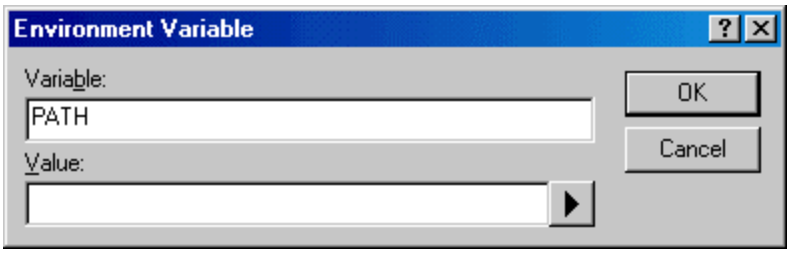


Step 4

The fourth and final step allows you to specify any environment variables that the phase requires.



To add a new environment variable click the [Add...] button (the **Environment Variable** dialog is displayed). Enter the variable name into the **Variable** field and the variable's value into the **Value** field and then click the **OK** button to add the new variable to the list. To modify an environment variable select the variable from the list and then click the **Modify** button. Make the required changes to the **Variable** and **Value** fields and then click the **OK** button to add the modified variable to the list. To remove environment variables select the variable that you want to remove from the list and then click the [Remove] button.



If the tool you are adding can display its output whilst the tool is running then use the [Read Output on Fly] option. This will display the tool output as each line of output happens. If this option is set to off then the HEW will store all output that is being displayed by the tool, and display it in the **Output** window when the tool has finished its operation. This can be a problem when the tool is running an operation that might take many minutes, as it is difficult to see the progress of the current execution.

Note:

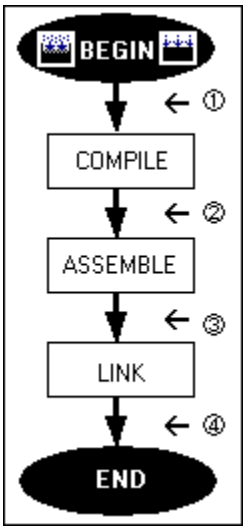
Using [Read Output on Fly] can cause problems when using certain tools on certain operating systems. If you are having problems with tools locking up or freezing in HEW then uncheck the [Read Output on Fly] option.

Click the **Finish** button to create the new phase. By default the new phase is added to the bottom of the **Build phase order** list in the **Build Order** tab of the **Build Phases** dialog.

3.3 Ordering Build Phases

In a standard build (shown in figure below), you could add a phase at four different positions: before the compiler, before the assembler, before the linker or after the linker.

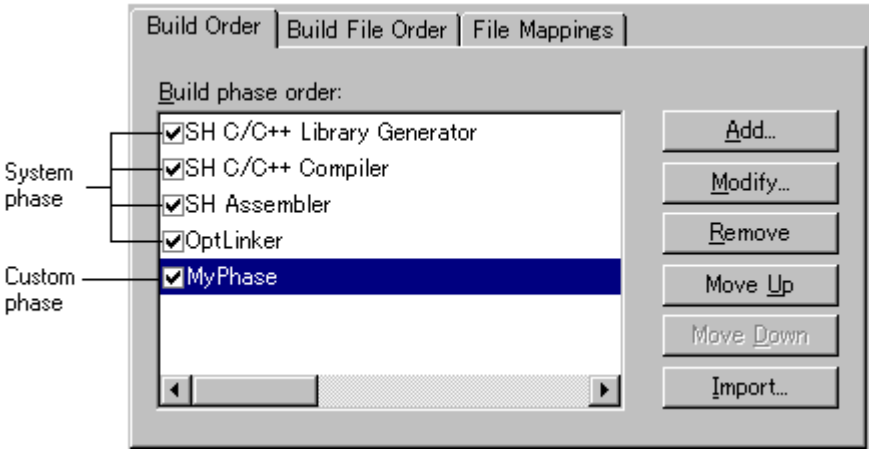
You may place your own custom phases or move system phases to any position in the build order. It is important to remember that if the output of your custom phase can be input into another phase then the phase order must be correct if the build is to behave as intended.



Select the [Build–>Build Phases...] menu option. The [Build Phases] dialog will be displayed. The build phase dialog provides facilities for ordering build phases via the [Build Phases] dialog. It has two tabs, which are concerned with the ordering of phases: [Build Order] and [Build File Order]. And then you can click [OK] button.

3.3.1 Build Order Tab

The **Build Order** tab displays the current order in which phases will be executed when the Build or Build All buttons is selected. The check boxes to the left of each phase indicates whether or not the phase is currently enabled. A phase can be toggled on/off by checking/unchecking its corresponding checkbox respectively.



To change the order of phases (system/custom) in a build or build all operation:

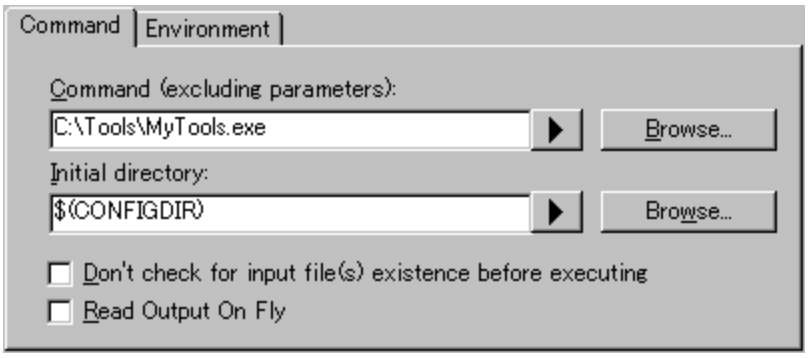
1. Select the phase to be moved and then click the [Move Up] and [Move Down] buttons to move the phase up and down respectively.
2. Click the **OK** button to set the new ordering.

To view the properties of a system phase:

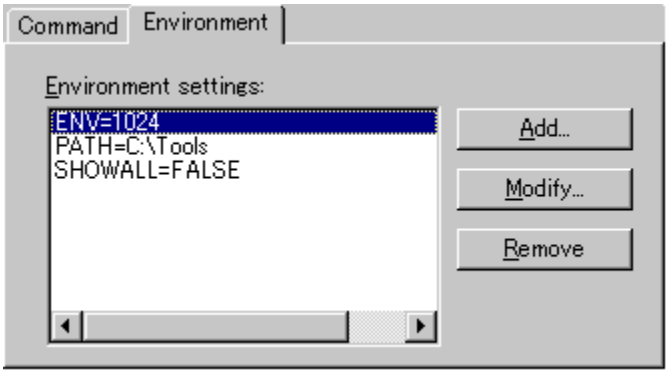
1. Select the system phase that you would like to view.
2. Click the [Modify...] button. The **Modify Phase** dialog will be invoked.
3. The **Command** tab shows general information about the phase. This may include copyright information, enhancements, bug fixes, user notes and so on.
4. Select the **Environment** tab to view the environment settings of the phase.
5. Click the **OK** button to close the dialog.

To modify a custom phase:

1. Select the custom phase that you would like to modify.
2. Click the [Modify...] button. The **Modify Phase** dialog will be invoked with the **Command** tab selected.



3. Change the contents of **Command** and **Initial directory** as appropriate.
4. Set the **Don't check for input file(s) existence before executing** checkbox if you don't want the HEW to abort the execution of the phase if any of the input files do not exist.
5. Select the **Environment** tab to edit the environment settings for the phase.



6. Use the [Add...], [Modify...] and [Remove] buttons to add, modify and remove environment variables respectively.
7. Click the **OK** button when all modifications have been made.

Note:

You can only change the environment of a system phase via the **Tools Administration** dialog.

To remove a custom phase:

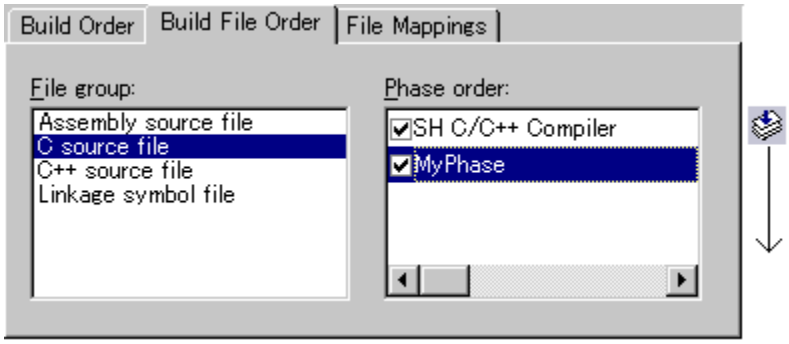
1. Select the phase to be removed and click the [Remove] button.
2. Click the **OK** button to confirm the new settings.

To import a custom phase:

1. Click the [Import...] button. The [Import Custom Phase] dialog is displayed, which allows you to browse to an existing project, from which you want to import a custom phase.
2. Choose the location of the project, from which you want to import a custom phase. Once selected, the [Import Phase] dialog is displayed, which lists the custom phases in the imported project.
3. Once you have decided which phase to import, highlight it in the list and click the **OK** button. The phase will then be added to the **Build Phases** dialog, at the bottom of the build order.

3.3.2 Build File Order Tab

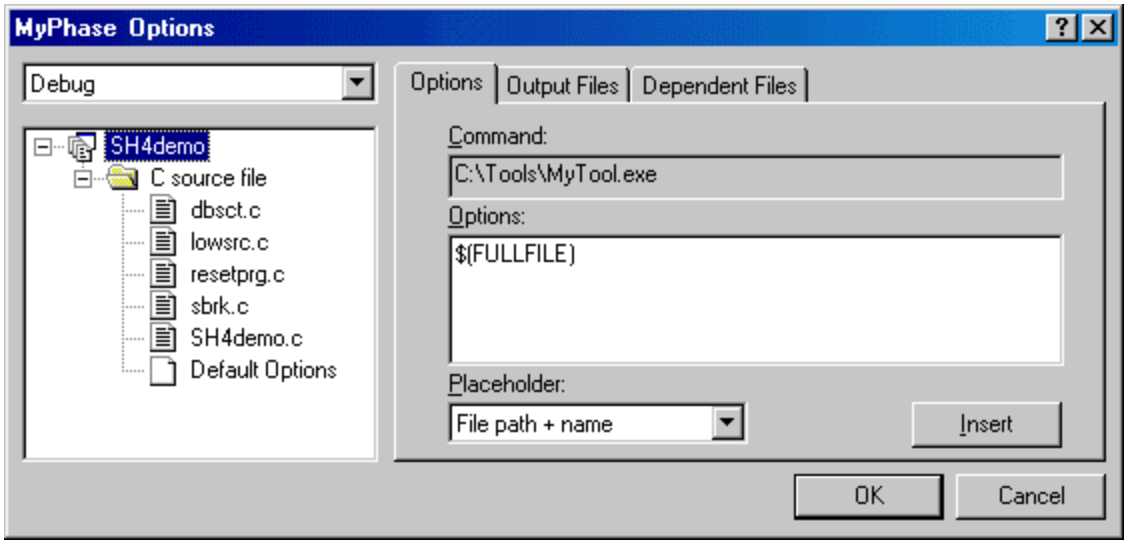
If you were to select a C source file from the Workspace window and then activate [Build->Build File] (or press ) you would expect the file to be compiled. Likewise, if you were to select an assembly source file from the workspace window and then activate [Build->Build File] you would expect the file to be assembled. The connection between file group and which phase(s) to execute is managed by the [Build File Order] tab of the [Build Phases] dialog. The list displays all of the current phases that will be executed when the build file operation is selected upon the file group shown in the [File group] list box. In figure below the “C source file” file group is selected and the “Compiler” and “MyPhase” phases are associated with it. Entries in the [Phase order] list, of the [Build File Order] tab, are added automatically as new entries are added to the [Build Order] tab.



For example, if you were to add a phase which takes C source files as input then this phase will be automatically added to the list of phases to execute when a build file operation is applied to a C source file. If you don’t want a certain phase to execute when [Build->Build File] is selected then clear the check box to the left of the phase name in the [Phase order] list.

3.4 Setting custom build phase options

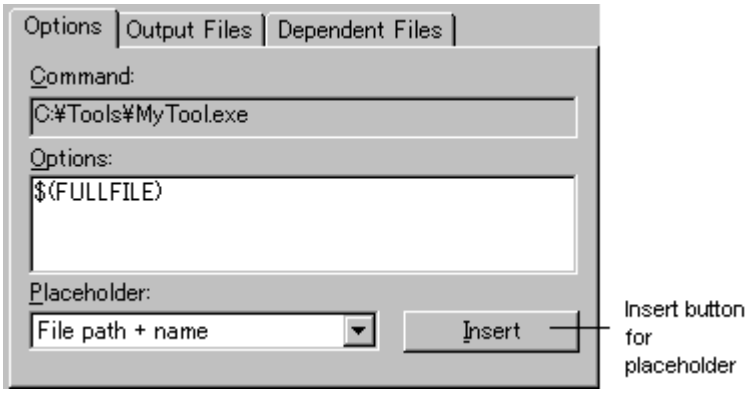
Once you have defined a custom phase, you will want to specify the command line options that should be used when it is executed. Each defined phase has a menu option in the **Build** menu. To specify options for that phase select it. The dialog that will be displayed depends on whether the custom phase selected was multiple or single (according to the selection of phase type when it was created in the New Build Phase wizard).



If the phase selected was multiple then a list of project files is displayed on the left-hand side of the dialog to enable you to specify the build options on a file by file basis. If the phase selected was single then there is no project file list displayed. In either case, the three tabs below are available. This is where you can set the options that you want to apply to the selected file(s). You can also choose which configurations are being viewed. In the configuration list, each configuration is listed along with a special entry named [Multiple configurations...]. If you select [Multiple configurations...] then the [Select project configurations to modify] dialog is displayed which allows you to select more than one configuration. This method is used throughout HEW for modifying multiple configurations at once.

3.4.1 Options Tab

This tab allows you to define the command line options that will be passed to the phase. The **Command** field displays the command that was entered when you defined the phase. Enter into the **Options** field the command line arguments that you would like to pass to the command. If you want to insert a placeholder, select the relevant placeholder from the **Placeholder** drop-down list and then click the **Insert** button. See Reference 4, Placeholders, for more information on placeholders.

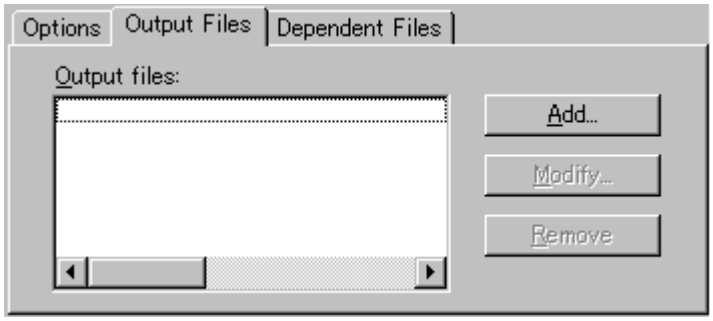


3.4.2 Output Files Tab

This tab is where you can specify the output file(s) that will be produced by the phase. Before each file is passed into this phase, the HEW checks that the output files are of a less recent date than the input file. If so, the phase will be executed for that file (i.e. input files have been modified since the output file or files were last produced). If the files are up-to-date then the phase will not be executed.

Note:

If no output files are specified, the phase will execute regardless.



To add an output file:

1. Click the [Add...] button. The **Add Output File** dialog will be invoked.
2. Enter the file path or browse to it using the **Browse** button.
3. Click the **OK** button to add this output file to the list.

To modify an output file:

1. Select the output file that you would like to modify.
2. Click the [Modify...] button. The **Modify Output File** dialog will be invoked.
3. Modify the fields as required.
4. Click the **OK** button to add the modified entry back to the list.

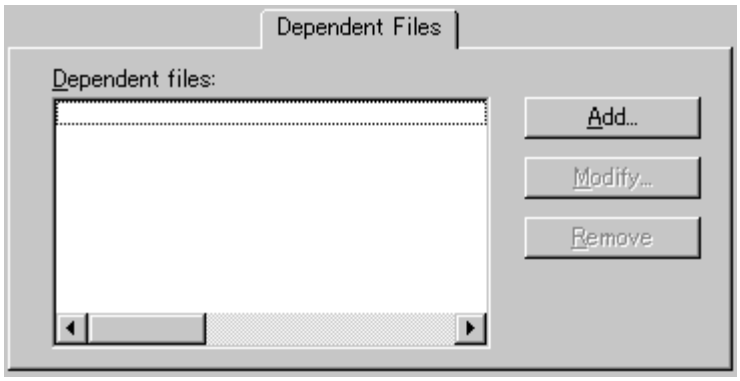
To remove an output file:

1. Select the output file that you would like to remove.
2. Click the [Remove] button.

An output file's path can include placeholders.

3.4.3 Dependent Files Tab

This tab is where you can specify the dependent files that are needed by the phase. Before each file is passed into this phase, the HEW checks that the dependent files are of a more recent date than the input file. If so, the phase will be executed for that file (i.e. dependent files have been modified since the input file(s) was last modified). If not, the phase will not be executed.



To add a dependent file:

1. Click the [Add...] button. The **Add Dependent File** dialog will be invoked.
2. Enter the file path or browse to it using the **Browse** button.
3. Click the **OK** button to add this output file to the list.

To modify a dependent file:

1. Select the dependent file that you would like to modify.
2. Click the [Modify...] button. The **Modify Dependent File** dialog will be invoked.
3. Modify the fields as required.
4. Click the **OK** button to add the modified entry back to the list.

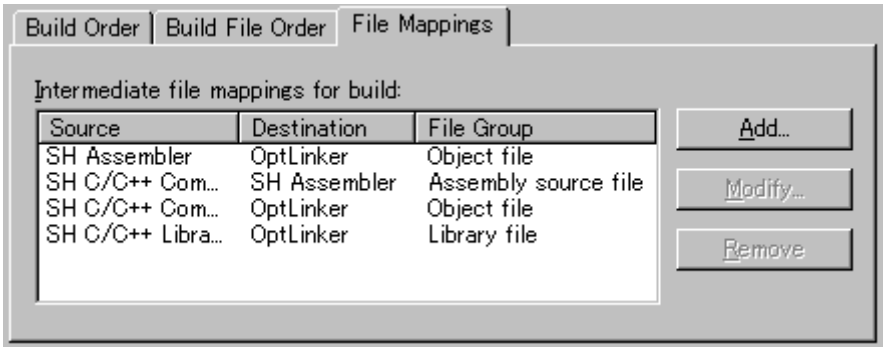
To remove a dependent file:

1. Select the dependent file that you would like to remove.
2. Click the [Remove] button.

A dependent file's path can include placeholders.

3.5 File mappings

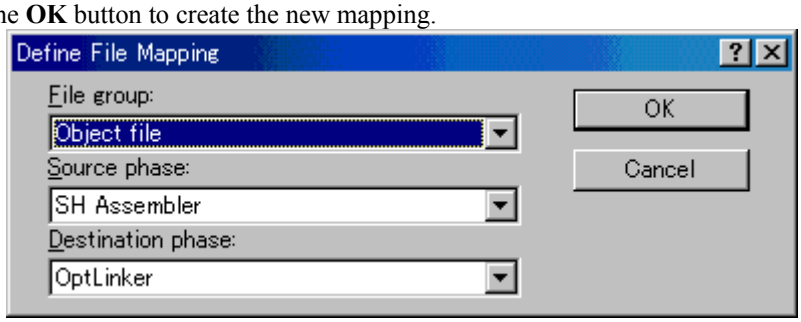
By default, the files input to a build phase are only taken from the project, i.e. all project files of the type specified in the **Select input file group** drop-down list on the **New Build Phase** dialog. Select the [Build–>Build Phases...] menu option. The [Build Phases] dialog will be displayed. If you would like a build phase to take files output from a previous build phase (these files are called intermediate files), then you must define this in the **File Mappings** tab of the **Build Phases** dialog.



A **File Mapping** means that you would like the output files of a certain type produced by one build phase (referred to as the **Source** phase) to another build phase (referred to as the **Destination** phase). Such intermediate files are then passed in addition to the project files.

To add a file mapping:

1. Click the [Add...] button. The **Define File Mapping** dialog will be invoked.
2. Select the source phase (i.e. the phase that generates the files) from the **Source phase** drop-down list.
3. Select the destination phase (i.e. the phase that takes these files) from the **Destination phase** drop-down list.
4. Click the **OK** button to create the new mapping.



To modify a file mapping:

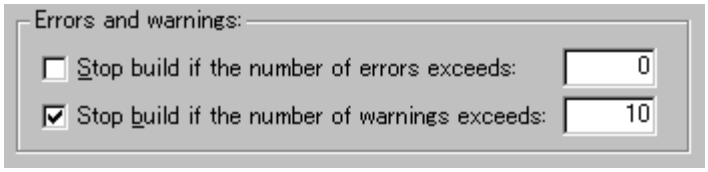
1. Select the mapping to be modified.
2. Click the [Modify...] button. The **Define File Mapping** dialog will be invoked.
3. Modify the options as necessary.
4. Click the **OK** button to commit the changes.

To remove a file mapping:

1. Select the mapping to be removed.
2. Click the [Remove] button.
3. Click the **OK** button to commit the changes.

3.6 Controlling the build

By default, the High-performance Embedded Workshop will execute all of the phases in a build and only stop if a fatal error is encountered. You can change this behavior by setting the controls on the **Build** tab of the [Options] dialog.



Select the [Setup–>Options...] menu option to display the [Options] and then select the **Build** tab. If you want to stop the build when a certain number of errors are exceeded then set the **Stop build if the no. of errors exceeds** checkbox and specify the error count limit in the field to the right. If you want to stop the build when a certain number of warnings are exceeded then set the **Stop build if the no. of warnings exceeds** checkbox and specify the warning count limit in the field to the right.

In addition to specifying error and warning count limits, the **Build** tab also allows you to request that the **Command line**, **Environment** and **Initial directory** of each execution should be displayed. Check the appropriate check boxes as necessary.

Note:

- Irrespective of what these controls are set to, the build will always halt if a fatal error is encountered.
- Note the following descriptions when you check the [Stop build if the number of errors exceeds] check box and specify a number to the controls.
    i. [Stop build if the number of errors exceeds] is not applied to a custom phase defined by a user. This is applied only to system phases such as the C/C++ Compiler, the Assembler and the OptLinker supplied from Renesas.
    ii. " number of errors" in the statement means the number of errors output from an execution of a tool. The total number of errors in more than one execution of a tool is irrelevant to [Stop build if the number of errors exceeds]. For example, the build stops when one execution of the C/C++ Compiler outputs more than the number specified. If the number of errors output from each compilation is no more than the specified number, the build does not stop even though the total number of errors in more than one compilation exceeds the specified number.
    iii. The build stops after the whole phase is done. For example, suppose you compile more than one file in a build. Even though the number of errors in the first compilation exceeds

the specified number, the HEW continues compilation of the other files and stops the build after finishing the whole compiler phase.

iv. When the number of errors in one execution of a tool exceeds the specified number, the number of error messages displayed on the “Output” window is the specified number plus one. A message saying that the number of error has exceeded the specified number is NOT displayed on the Output window.

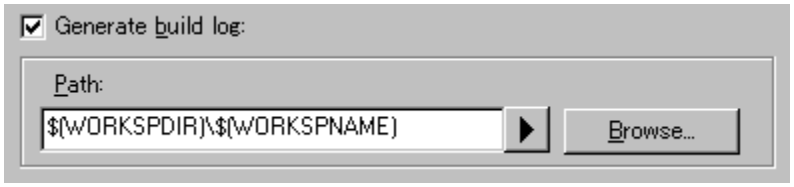
The same things as [Stop build if the number of errors exceeds] shown above hold also in [Stop build if the number of warnings exceeds]. There is no correlation between [Stop build if the number of errors exceeds] and [Stop build if the number of warnings exceeds]. They are independent.

3.7 Logging build output

The HEW allows you to write the results of each build to file.

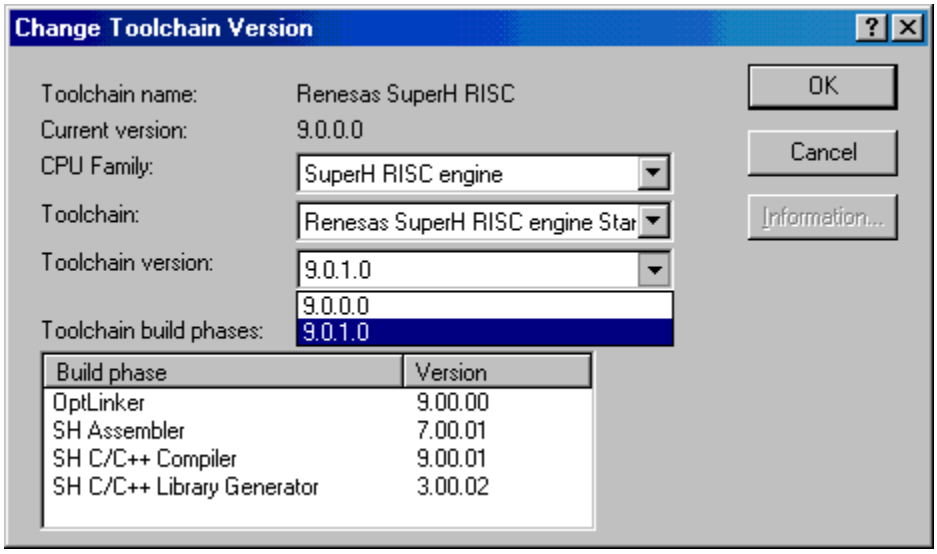
To specify a log file:

1. Select the [Setup–>Customize...] menu option. The [Customize] dialog will be displayed. Select the **Log** tab.
2. Set the **Generate build log** checkbox.
3. Enter the full path of the log file into the **Path** field, or browse to it graphically by clicking the [Browse...] button.
4. Click the **OK** button to confirm the new log file settings.



3.8 Changing Toolchain Version

If two or more versions of the same toolchain are registered in the HEW, you can choose a version of the toolchain on the [Change Toolchain Version] dialog. To invoke the dialog, select the [Tools–>Change Toolchain Version...] menu option. Choose one of the versions from the **Toolchain version** drop-down list and click the **OK** button to enforce your choice.



To show information about toolchain components, select a tool from the [Toolchain build phase] list on the [Change Toolchain Version] dialog, and click the [Information...] button. A tool information dialog will show you information about the tool. Click the **Close** button to close the dialog.

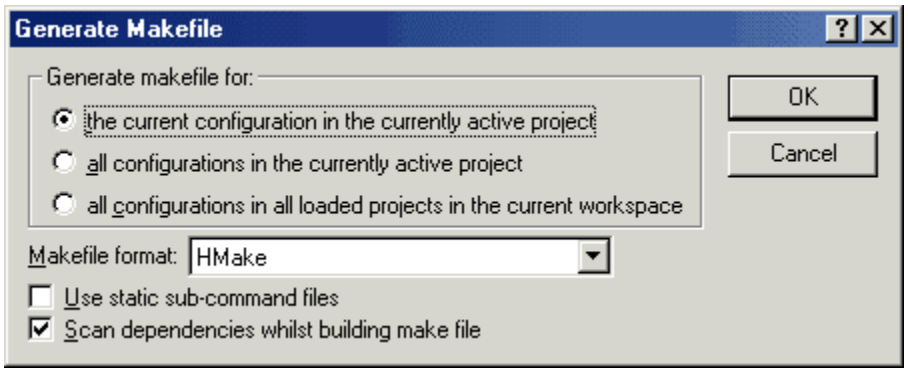
3.9 Generating a makefile

The HEW allows you to generate a makefile, which can be used to build parts of your workspace in the DOS Command Prompt without HEW. This is particularly useful if you want to version control an entire build, including the make components.

This version of HEW also supports the integration of the generated makefile and tools into the HEW framework. This allows you to click build menu or toolbar button and then launch the makefile rather than the normal internal HEW build process. For further information, see section 3.10, Using a makefile inside the HEW system.

To generate a makefile:

1. Ensure that the project that you want to generate a makefile for is the current project.
2. Ensure that the build configuration that you want to build the project with is the current configuration.
3. Select the [Build→Generate Makefile...] menu option.
4. Once this menu has been selected, a dialog is displayed, which asks the user what parts of the workspace need to be added to the makefile. Select the radio button that is relevant for your makefile.
5. Ensure you have selected the correct makefile format. HEW is capable of generating GNUMake, HMake and NMake compatible files.
6. The use static sub-command files check box will generate separate command files in the make destination directory. This is different to the normal style where the sub-command files are generated by the make tool. In the case of the GNUMake file format this must be switched on.
7. Checking [Scan dependencies whilst building makefile] will force a dependency scan to ensure the makefile creation is up to date.
8. Click [OK].



The HEW will create a subdirectory called 'make' within the current workspace directory and then generate the makefile into it. It is named after the selection and has a `.MAK` extension for example the current project and configuration (e.g. `PROJECT_DEBUG.MAK`). The executable `HMAKE.EXE`, located in the HEW installation directory, is provided for you to execute the makefiles generated by the HEW. It is not intended to execute makefiles that have been modified by the user. For further details on HMAKE, see Reference 12, HMAKE User Guide.

Note:

- If the name of the HEW installation directory includes a space, the GNU Make command will not work correctly when GNU Make is selected as the makefile format with the makefile generating function.
- When a makefile is generated with a sub-command specified, the HEW outputs the directory of sub-command files as an absolute path. If these sub-command files are moved to another directory, the make commands will not work correctly because the make commands cannot refer to the sub-command files.

To execute a makefile:

1. Open a DOS Command Prompt window and change to the 'make' directory where the makefile was generated.
2. Execute `HMAKE`. Its command line is `HMAKE.EXE <makefile>`.

Note:

The degree of portability of a generated makefile is entirely dependent upon how portable the project itself is. For example, any compiler options that include full paths to an output directory or include file directory will mean that, when given to another user with a different installation, the build will probably fail. In general use placeholders wherever possible – using a full, specific path should be avoided when possible.

3.10 Using makefiles to build inside HEW

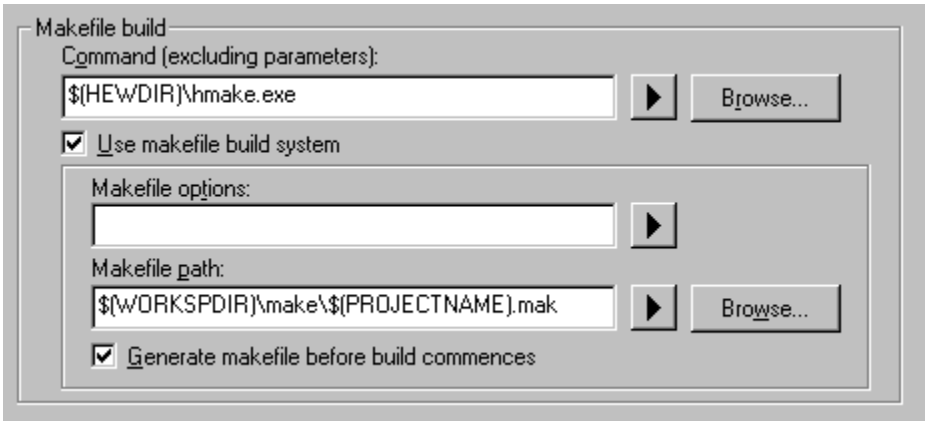
The HEW also allows you to configure the internal build to use a particular make file rather than the internal HEW system when build is clicked.

The standard HEW integrated build should be fine for all of your building requirements. However in some cases you may feel the need to use makefiles rather than relying on the HEW internal make system. The techniques to do this are described below:

To set-up the internal makefile execution:

1. Create a base HEW workspace in which to launch your makefile in. It makes sense if this is the same toolchain as the one to be used in your makefile. Although this is not essential.
2. Click on [Setup->Options...].
3. Click the [Build] tab.
4. Select the makefile command, which should be used. By default this is set to the Renesas make tool shipped with HEW. This is called HMAKE.exe and is located in the HEW installation directory "$(HEWDIR)".
5. Click the "Use makefile build system" check box. This tells the HEW that when the build button is clicked that makefile should be executed rather than the internal build.
6. Setup the makefile path. This defaults to the location which HEW will generate makefiles to. This item is configuration based so you can have a different makefile for each configuration.

   Note: You do not need to use a makefile generated by HEW. It can be any format as long as your make tool supports it.
7. Setup the makefile options. HMAKE allows the user to specify project or configuration selection options so that different builds can be launched from the same file. Again this item is configuration based so you can have a different makefile options for each configuration.
8. The final option is the generate make file before build. This will launch the default make file creation utility before the makefile is executed. This means the makefile will always be up to date with the HEW project system.
9. Click OK to save the changes.



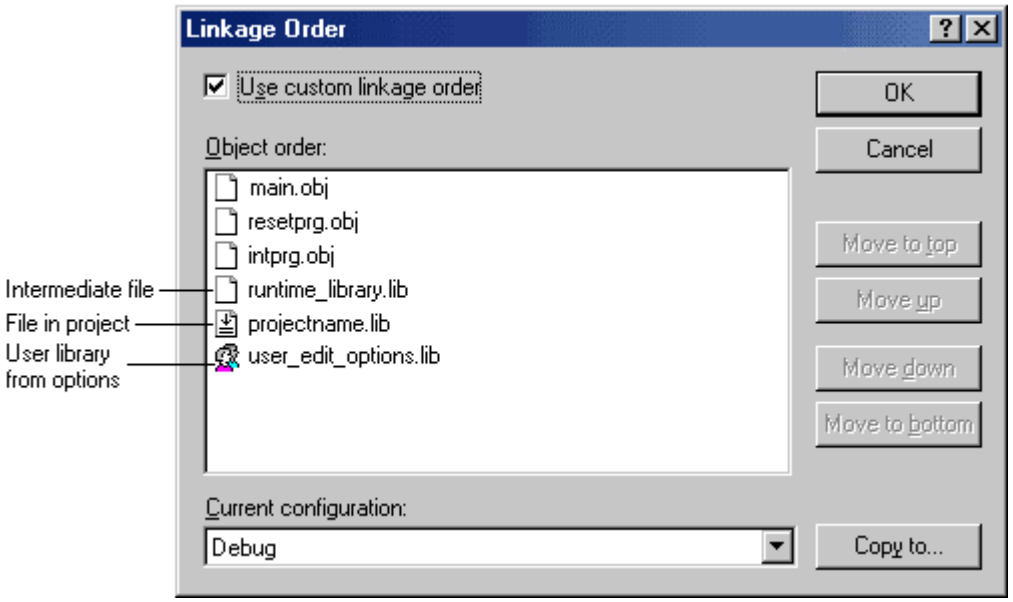
Once this system is setup the build button, menu and keyboard shortcuts are linked to the makefile tool execution. All output is directed to the output window as in the case of the normal build. If you are using a toolchain supported by HEW errors and warnings can be double-clicked to jump to the source files. The help link should also be supported. Note when using the HMAKE.exe system the build all button will pass a command to HMAKE to force a re-build all operation.

3.11 Customizing the HEW linkage order

Object modules are linked as alphabetical order in HEW default. You can specify the linkage order, if you wish to.

To switch on manual linkage order facilities:

1. Click the [Build->Linkage Order...] menu item.
2. On the [Linkage Order] dialog displayed click the [Use custom linkage order] check box.
3. Then you can move the objects into which ever order you need. Simply select the module and click the [Move to top], [Move up], [Move down] and [Move to bottom] buttons to position it in the desired location.
4. Each module has a different icon depending on where it originated from. This is shown below:
5. Click OK to verify and save the settings.



When you are using multiple configurations it is likely that the linkage order will be very similar. To do this effectively you can copy the current settings in the dialog to other configurations. This is described below:

To copy the linkage order from one configuration to another:

1. Click the [Build->Linkage Order...] menu item.
2. Select the configuration you wish to copy in the current configuration list. This defaults to the currently loaded configuration.
3. Click the [Copy to...] button this displays the [Select Configuration To Copy To] dialog and asks you which of the configurations in the current project you wish to copy the current linkage order to. Select a configuration and click OK.
4. Click OK to verify and save the changes.

4. Using the Editor

4.1 Editor window

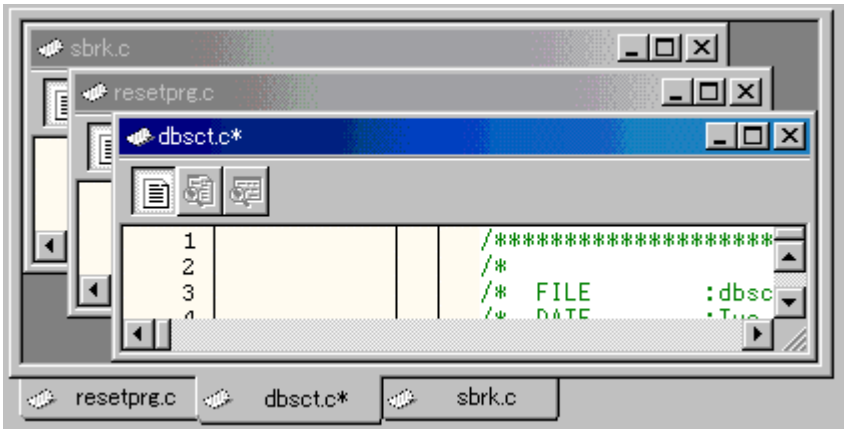
The editor window contains the file windows that are being viewed or edited.

The title bar of the active window will appear a different color from that of the others (“dbsct.c” is the active window in figure bellow). All text operations such as typing, pasting text and so forth only affect the active window.

To switch to another source file window (i.e. to make some other window the active window) there are a number of methods:

- Click on it if it is visible, OR
- Press CTRL+TAB or SHIFT+CTRL+TAB to cycle through the windows one after another, OR
- Select the window by name from the **Window** menu, OR
- Select its tab at the bottom of the editor window.

When a file has been edited, an asterisk (*) is appended to the window’s title bar. The asterisk remains there until the file is saved. The asterisk is also removed if all of the edited changes are undone in the current window.

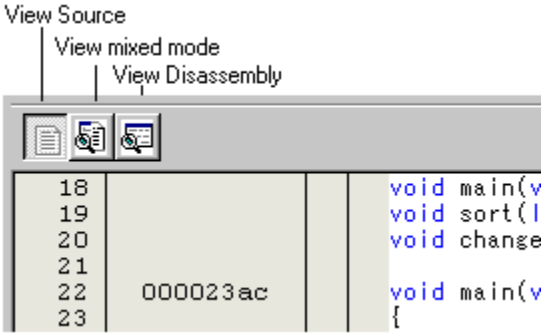


Selecting text in the editor:

It is possible to select text in the same manner as all editors. However to access column selection hold down the ALT key while you are selecting the text with the mouse. This changes the selection technique from line to column selection.

4.2 Integrated disassembly view in the editor

The HEW editor in version 4.0 onwards has been enhanced to include an integrated disassembly view. This integrated view has a toolbar which allows the switching of mode. When each mode is available it is possible to click the button and change to the new view. This toolbar is shown below:



Three different modes are possible these are listed below:

Mode	Function
View Source	This mode is the standard HEW editor. It allows source file editing and keywords are highlighted correctly if you are viewing source files. In this view you can view line numbers, source addresses, coverage, breakpoints and bookmarks.
View Mixed mode	The mixed mode facility in a source file is different to the disassembly window mixed mode. Instead of showing the continuous disassembly it shows the disassembly that is related to each line of source code. This view cannot be edited and is only available when a debugger target is attached to the session. This view allows breakpoints, bookmarks and line numbers to be viewed.
View Disassembly	The disassembly mode shows the true continuous disassembly code in address order. This is the same as clicking the [View->Disassembly] menu item. Labels can be viewed in this view as well as addresses and disassembled code.

Note:

It is not possible to switch from the source window to the mixed display under the following conditions.

1. The target is not connected to the current session.
2. No download modules have been downloaded.
3. No debug information is available for the current project.
4. The currently displayed file has been edited and the changes not saved.

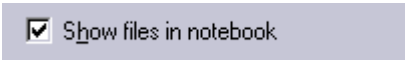
4.3 Working with Multiple Files

The file area is where you will work with the files of your project. The editor allows you to have many files open at one time, to switch between them, to arrange them in different configurations and to edit them in whichever order you want to. The operations that you can perform upon the windows are typical of most Windows® applications and they can be found under the **Window** menu:

Menu Option	Operation
[Window→Cascade]	Arrange all open windows so that they overlap, with the top left of each Editor window visible.
[Window→Tile Horizontally]	Arrange all open windows horizontally so that they occupy the entire Editor window, without any overlaps.
[Window→Tile Vertically]	Arrange all open windows vertically so that they occupy the entire Editor window, without any overlaps.
[Window→Arrange Icons]	Line up all minimized windows at the bottom of the Editor window.
[Window→Close All]	Close all open Editor windows.

The files within the editor can be displayed in a 'notebook' style. This means that each file has a separate tab associated with it to aid in navigating between files.

To show files in a notebook style:



1. Select the [Setup–>Options...] menu option. The [Options] dialog will be displayed. Select the **Editor** tab.
2. Set the **Show files in notebook** checkbox as appropriate.
3. Click the **OK** button for the new settings to take effect.

4.4 Standard File Operations

4.4.1 Creating a New File

To create a new editing window:

- Click the New File toolbar button  , **OR**
- Press CTRL+N, **OR**
- Select the [File→New] menu option.

The new window will be given an arbitrary name by default. You can provide a new name when you save the file.

4.4.2 Saving a File

To save the contents of an editing window:

1. Ensure that the window, whose contents you want to save, is the active window.
2. Select one of the following operations:

- Click the Save File toolbar button  , **OR**
- Press CTRL+S, **OR**
- Select the [File→Save] menu option.

3. If the file has not been saved before, a **File Save** dialog will be displayed. Enter a filename, specify a directory and then click the **OK** button to create the file with the name given in the directory specified. If the file has been saved before then the file will be updated (no dialog will be displayed).

To save the contents of an editing window under a new name:

1. Ensure that the window, whose contents you want to save, is the active window.
2. Select the [File→Save As] menu option.
3. A **File Save** dialog will be displayed. Enter a filename, specify a directory and then click the **OK** button to create the file with the name given, in the directory specified.

To save the contents of every open editor window:

1. Select one of the following operations:
   - Click the Save All toolbar button  , **OR**
   - Select the [File→Save All] menu option.
2. If any of the files has not been saved before, a **File Save** dialog will be displayed. Enter a filename, specify a directory and then click the **OK** button to create the file with the name given in the directory specified. If any of the files have been saved before, then that file will be updated (no dialog will be displayed).

4.4.3 Opening a File

To open a file:

1. Select one of the following operations:
   - Click the File Open toolbar button  , **OR**
   - Press CTRL+O, **OR**
   - Select the [File→Open] menu option.
2. A **File Open** dialog will be displayed. Use the directory browser to navigate to the directory in which the file you want to open is located. Use the **Files of Type** combo box to select the type of file you want to open (or set it to **All Files (*.*)** to see every file in a directory).
3. Once you have located the file, select it and click **Open**.

You can also open a file via the **Projects** tab of the **Workspace** window. Either double click the file you want to open, or select it, right-click and then choose the [Open <file>] menu option on the pop-up menu (where <file> is the name of the file selected).

The High-performance Embedded Workshop keeps track of the last five files that you have opened and adds them to the File menu under the [File→Recent Files] sub-menu. This gives you a shortcut to opening the last 4 files that you have used.

To open a recently used file:

- Select the [File→Recent Files] menu option and from this sub-menu select the desired file.

4.4.4 Closing Files

To close an individual file:

- Double-click the **Editor** window's system menu (located at the top left of each window), **OR**
- Click on the **Editor** window's system menu (located at the top left of each window) and select the **Close** menu option, **OR**
- Ensure that the window that you want to close is the active window and then press CTRL+F4, **OR**
- Ensure that the window that you want to close is the active window and then select the [File→Close] menu option.

To close all files:

- Select the [Window→Close All] menu option.

4.5 Editing a File

The HEW editor's standard editing functionality is available through the usual methods (i.e. the menu, toolbar and keyboard shortcuts) and is additionally supported via a pop-up menu that is local to each **Editor** window. Right-click in an open **Editor** window to invoke the pop-up menu.

The table below outlines the basic operations that are provided by the editor.

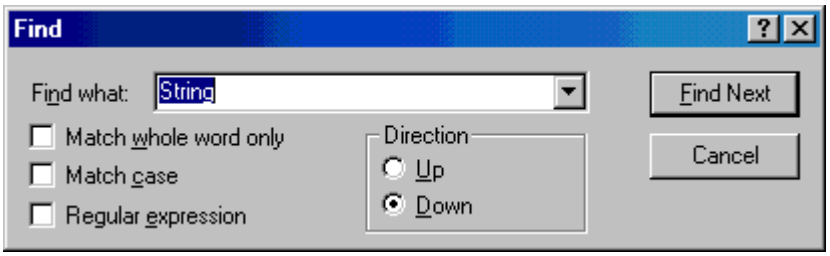
Operation	Effect	Action
Undo	Reverses the last editing operation	Select [Edit→Undo] Press CTRL+Z
Redo	Repeats the last undone editing operation	Select [Edit→Redo] Press CTRL+Y
Cut	Removes highlighted text and places it on the Windows® clipboard	Click the Cut toolbar button  Select [Edit→Cut] Select [Cut] from the pop-up menu Press CTRL+X
Copy	Places a copy of the highlighted text into the Windows® clipboard	Click the Copy toolbar button  Select [Edit→Copy] Select [Copy] from the pop-up menu Press CTRL+C
Paste	Copies the contents of the Windows® clipboard into the active window at the position of the insertion cursor	Click the Paste toolbar button  Select [Edit→Paste] Select [Paste] from the pop-up menu Press CTRL+V
Clear	Removes highlighted text (it is not copied to the Windows® clipboard)	Select [Edit→Clear] Press Delete
Select All	Selects (i.e. highlights) the entire contents of the active window	Select [Edit→Select All] Press CTRL+A

4.6 Searching and Navigating through Files

4.6.1 Finding Text

To search for text in the current file:

1. Ensure that the window, whose contents you want to search, is the active window.
2. Position the insertion cursor at the point from which you want to start your search.
3. Select one of the following operations to display the **Find** dialog:
   - Click the Find toolbar button  , **OR**
   - Press CTRL+F, **OR**
   - Select the [Edit→Find] menu option, **OR**
   - Select the **Find** option from the **Editor** window's pop-up menu.

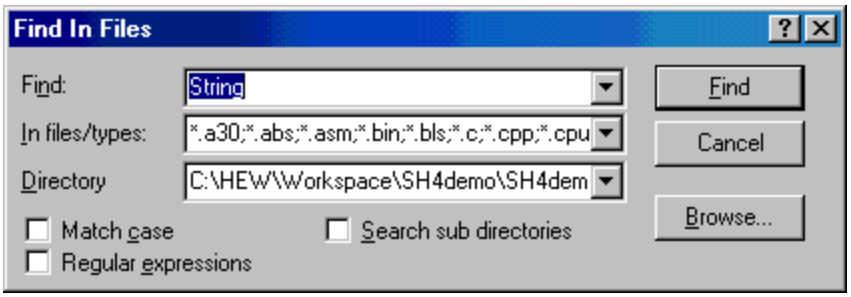


4. Enter the text that you want to search for into the **Find What** field, or select a previous search string from the drop-down list box. If you select text before invoking the find operation, the selected text will be automatically placed into the **Find What** field.
5. If you would like to search for character string as a whole word then click the **Match Whole Word Only** checkbox. When this option is not selected, the search will be for any string that is matched by the search string.
6. If you would like your search to be case-sensitive (i.e. to distinguish between upper and lower case letters) then check the **Match Case** checkbox.
7. If your search string uses regular expressions then check the **Regular Expressions** checkbox. See Reference 3, Regular Expressions, for further information.
8. The **Direction** radio buttons allow you to select the direction of the search. Selecting **Down** means that the search will be performed from the insertion cursor towards the bottom of the file. Selecting **Up** means that the search will be performed from the insertion cursor towards the top of the file.
9. Click the **Find Next** button to begin the search. Click the **Cancel** button to stop the **Find** action.

4.6.2 Finding Text in Multiple Files

To search for text in multiple files:

1. Select one of the following operations to display the **Find In Files** dialog:
   - Click the Find In Files toolbar button  , **OR**
   - Select the [Edit→Find in Files] menu option, **OR**
   - Select **Find in Files** from the **Editor** window's pop-up menu.



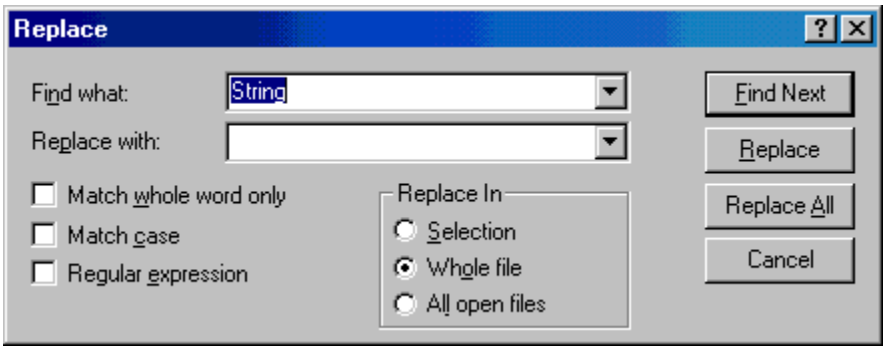
2. Enter the text that you want to search for into the **Find** field, or select a previous search string from the drop-down list box. If you selected text before invoking the find operation, the selected text will be automatically placed into the **Find** field.
3. Enter the file extensions of the files you would like to search into the **In Files/Types** field. If several extensions are specified, be sure to separate them with a comma (e.g. `*.C, *.H`).
4. Enter the directory that contains the files to search, into the **Directory** field. Alternatively you may browse to the desired directory graphically if you click the **Browse** button.
5. If you would like to search the directory specified and all directories below it then check the **Search Sub-Directories** checkbox. If you just want to search the single directory specified in **Directory** field then ensure that this checkbox is not checked.
6. If you would like your search to be case-sensitive (i.e. to distinguish between upper and lower case letters) then check the **Match Case** checkbox.
7. If your search string uses regular expressions then check the **Regular Expressions** checkbox. See Reference 3, Regular Expressions, for further information.
8. Click **Find** to begin the search. Any matches found will be displayed in the **Find In Files** tab of the **Output** window. To stop a **Find In Files** action once it is under way, select the [Edit→Stop Find in Files] menu option. Once the **Find In Files** operation is complete, you may jump to an instance of the search string by double-clicking on the desired entry in the **Output** window.

4.6.3 Replacing Text

Replacing text is similar to finding text, as discussed in the previous section. The difference is that when the text is found you have the option to replace it with other text.

To replace text in a file:

1. Ensure that the window, whose contents you want to replace, is the active window.
2. Position the insertion cursor at the point from which you want to start your search.
3. Select the [Edit→Replace] menu option or select **Replace** from the **Editor** window's pop-up menu. A **Replace** dialog will be displayed.

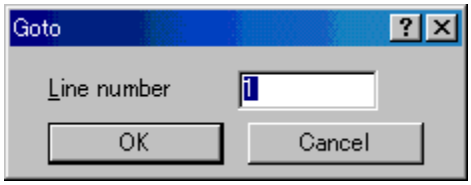


4. Enter the text that you want to search for into the **Find What** field, or select a previous search string from the drop-down list box. If you select text before invoking the replace operation, the selected text will be automatically placed into the **Find What** field.
5. Enter the text that you want to replace the search string with, or select a previous replace string from the drop-down list box.
6. If you would like to search for character string as a whole word then click the **Match Whole Word Only** checkbox. When this option is not selected, the search will be for any string that is matched by the search string.
7. If you would like your search to be case-sensitive (i.e. to distinguish between upper and lower case letters) then check the **Match Case** checkbox.
8. If your search string uses regular expressions then check the **Regular Expressions** checkbox. See Reference 3, Regular Expressions, for further information.
9. If you clicked the Find Next button, the editor will search for the first occurrence of the search string. Click the **Replace** button if you want to replace it. Click **Replace All** button to replace all occurrences or click the **Cancel** button to stop the replace action. If you select **Selection** in the **Replace In** field, the replace action will be performed in the range of the selected text. If you select **Whole File**, the replace action will be performed on the whole file. If you select **All Open Files**, all files that are currently open in the editor will have the replace operation carried out on them.

4.6.4 Jumping to a Specified Line

To jump to a line in a file:

1. Ensure that the window, whose contents you want to replace, is the active window.
2. Select one of the following operations to display the Goto dialog:
   - Press CTRL+G, **OR**
   - Select the [Edit→Goto Line] menu option, **OR**
   - Select **Goto Line** from the **Editor** window's pop-up menu.



3. Enter the number of the line that you want to jump to into the **Line Number** box and then click the **OK** button.
4. The insertion cursor will be placed at the start of the line number specified.

4.7 Bookmarks

When working with many large files at a time, it can become difficult to locate specific lines or areas of interest. Bookmarks enable you to specify lines that you want to jump back to at a subsequent time. One example of its use is in a large C file where you may want to set a bookmark on each function definition. Once a bookmark has been set, it exists until it is removed or the file is closed.

To set a bookmark:

1. Place the insertion cursor on the line to mark.
2. Select one of the following operations:
   - Click the Toggle Bookmark toolbar button  , **OR**
   - Press CTRL+F2, **OR**
   - Select the [Edit→Bookmarks→Toggle Bookmark] menu option, **OR**
   - Right-click and select the [Bookmarks→Toggle Bookmark] menu option from the pop-up menu.

A bookmark icon  will be placed on the bookmarked line, to indicate that it is an active bookmark.

To remove a bookmark:

1. Place the insertion cursor on the marked line.
2. Select one of the following operations:
   - Click the Toggle Bookmark toolbar button  , **OR**
   - Press CTRL+F2, **OR**
   - Select the [Edit→Bookmarks→Toggle Bookmark] menu option, **OR**
   - Right-click and select the [Bookmarks→Toggle Bookmark] menu option from the pop-up menu.

The bookmark icon will be removed from the line.

To jump to the next bookmark in a file:

- Click the Next Bookmark toolbar button  , **OR**
- Press F2, **OR**
- Select the [Edit→Bookmarks→Next Bookmark] menu option, **OR**
- Right-click and select the [Bookmarks→Next Bookmark] menu option from the pop-up menu.

To jump to the previous bookmark in a file:

- Click the Previous Bookmark toolbar button  , **OR**
- Press SHIFT+F2, **OR**
- Select the [Edit→Bookmarks→Previous Bookmark] menu option, **OR**
- Right-click and select the [Bookmarks→Previous Bookmark] menu option from the pop-up menu.

To remove all bookmarks:

- Click the Clear All Bookmarks toolbar button  , **OR**
- Select the [Edit→Bookmarks→Clear All Bookmarks] menu option, **OR**
- Right-click and select the [Bookmarks→Clear All Bookmarks] menu option from the pop-up menu.

4.8 Printing a File

To print a file:

Ensure that the window, whose contents you want to print, is the active window.

Select one of the following operations:

- Click the Print toolbar button  , **OR**
- Press CTRL+P, **OR**
- Select the [File→Print] menu option.

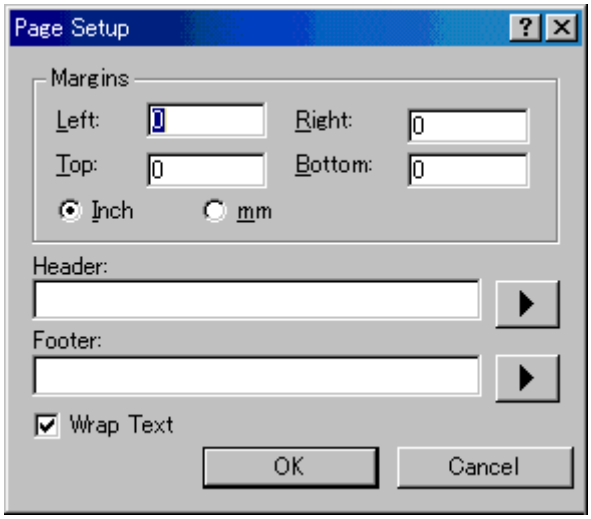
4.9 Configuring Text Layout

4.9.1 Page Set-up

When you print a file from the HEW editor, the settings in the print dialog affect the way the file is printed (e.g. double or single sided). Control over how the text is formatted on the page can also be controlled via the **Page Setup** option. This allows you to specify the margins (top, bottom, left and right) of your printouts. It is often necessary to set this because some printers cannot print to the edges of an A4 page. Furthermore, some users have their own layout requirements (e.g. a large left-hand margin so that code can be placed in an A4 binder).

To set up the page margins:

1. Select the [File->Page Setup...] menu option. The **Page Setup** dialog will be invoked.
2. Enter the width of the margins required into the **Left**, **Right**, **Top** and **Bottom** fields.
3. Set the **Inch** or **mm** option accordingly.
4. Click the **OK** button for the new settings to take effect.



To set up the header and footer information:

1. Select the [File->Page Setup...] menu option. The **Page Setup** dialog will be invoked.
2. Enter into the **Header** and **Footer** edit fields the text required to be displayed. All normal placeholders are available along with page numbering, text justification and date fields. These are all expanded before the page is to be printed.
3. Click the **OK** button for the new settings to take effect.

To set up print wrapping

1. Select [File->Page Setup...]. The "Page Setup" dialog will be invoked.
2. Click the wrap text check box. This switches on the wrap text facility when printing so no text is truncated and everything is visible.
3. Click "OK" for the new settings to take effect.

4.9.2 Changing Tabs

When the TAB key is pressed in the editor a tab character is usually stored in the file. However, sometimes it is preferable to store spaces instead. The representation of tab characters can be controlled via the [Options] dialog.

To change tab size:

1. Select the [Setup–>Options...] menu option. The [Options] dialog will be displayed. Select the [Editor] tab.
2. Enter into **Tab Size** the number of desired tabs.
3. Click the **OK** button for the new tab settings to take effect.

To use spaces as tabs:

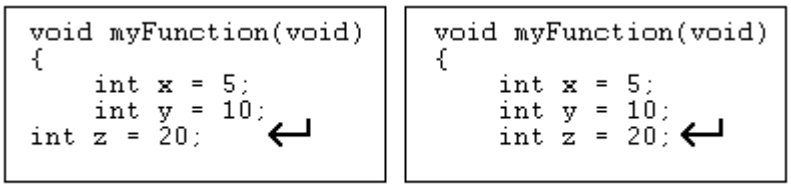
1. Select the [Setup–>Options...] menu option. The [Options] dialog will be displayed. Select the [Editor] tab.
2. Set the **Use spaces as tabs** checkbox as appropriate.
3. Click the **OK** button for the new tab settings to take effect.

4.9.3 Auto Indentation

When you press RETURN in the editor, the insertion cursor will move to the next line down, at the first column (i.e. against the left-hand side of the window). **Auto Indentation** is a feature which, when RETURN is pressed, places the insertion cursor on the next line (as before) but under the first non-whitespace character of the previous line. This enables you to type neat C/C++ or assembler code faster as you don't have to type leading spaces or tabs yourself.

The figure below illustrates two examples. The first shows the effect of pressing RETURN when the **Auto Indentation** feature is disabled – the insertion cursor returns to the left-hand side of the window on the next line. When the '`int z = 20`' line is typed, it is not aligned with the previous two lines. The second

example shows the effect of pressing RETURN when **Auto Indentation** is enabled – the insertion cursor drops underneath the '`i`' of the '`int`' word on the previous line. Now, when the '`int z = 20`' line is typed, it is automatically aligned (i.e. automatically indented).



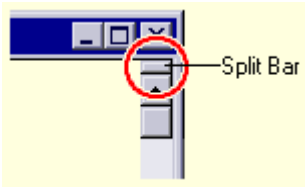
(i) Auto Indentation Disabled (ii) Auto Indentation Enabled

To enable/disable Auto Indentation:

1. Select the [Setup–>Options...] menu option. The **Setup Options** dialog will be displayed. Select the **Editor** tab.
2. Set the **Enable Auto Indentation** checkbox accordingly.
3. Click the **OK** button for the new settings to take effect.

4.10 Splitting a Window

The HEW editor allows you to split a text window into two. The split bar button is located just underneath the maximize button at the top right-hand corner of any text window (as shown below).



To split a window:

- Double-click the split bar button to split the window in half, or click on the split bar button, keep the button pressed, drag the mouse down and then release the mouse button at the point you want to split the window.

To adjust the position of the split bar:

- Click on the split bar itself, keep the button pressed then move the bar to the new position and then release the button.

To remove the split bar:

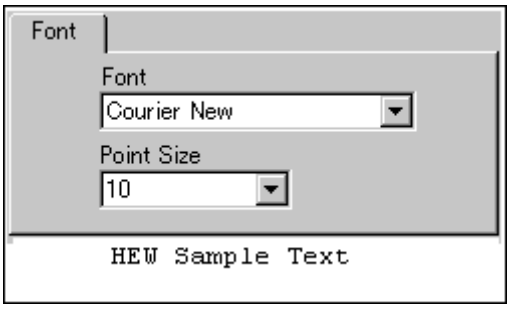
- Double-click on the split bar or move the split bar to the top or bottom of the window.

4.11 Changing the Editor Font

The High-performance Embedded Workshop allows you to specify the font to be used in its internal editor. All **Editor** windows, regardless of the file type, use the same font.

To change the editor font:

1. Select the [Setup–>Format Views...] menu option. The [Format Views] dialog will be displayed.
2. Click on the source icon.
3. The font tab should be available on the right of the format views dialog. Select the font type and size in the drop lists. When this is being modified the sample text below shows what the font will look like.
4. Click the **OK** button for the new font settings to take effect.



4.12 Syntax Coloring

4.12.1 Syntax coloring

To enhance code readability, the HEW editor can display specific strings (i.e. keywords) in different colors. For instance, C source code comments could be shown in green, and C types (e.g. `int`) could be shown in blue.

The coloring method used can be specified on a file group by file group basis. For example, you can define different color schemes for C source files, text files, map files or even your own files.

Note:

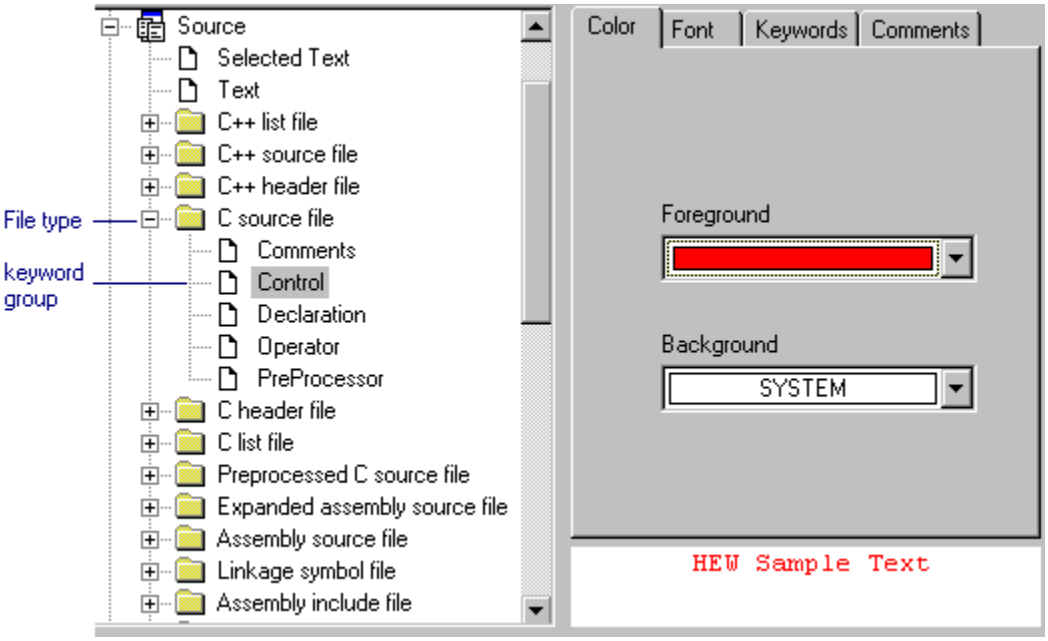
When you create a new file, syntax coloring will not be active, as a new file does not initially have an extension (new files are named arbitrarily by the editor without an extension). In order to activate syntax coloring, you must save the new file with a name and extension that the HEW recognizes. See the File extensions and file groups topic for information on file extensions.

4.12.2 Changing text colors

To change existing colors:

1. Select the [Setup–>Format Views...] menu option. The **Format Views** dialog will be displayed. Select the view you are interested in changing the font for. If it is the editor expand the source view in the tree on the left hand side of the dialog.
2. Select the file type for which you want to edit syntax coloring from the File group list and then expand and select it.

3. Select the keyword group you are interested in. The tabs on the right side of the dialog change depending on the selection.
4. Select the [Color] tab.
5. Modify the **Foreground** and **Background** color lists as desired. The **System** color refers to the current window foreground and background settings in Control Panel.
6. Click the **OK** button for the new color settings to take effect.



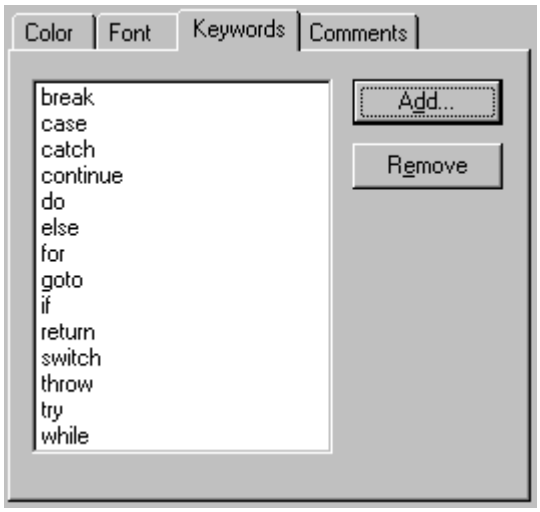
4.12.3 Creating new keywords

To create new keyword groups:

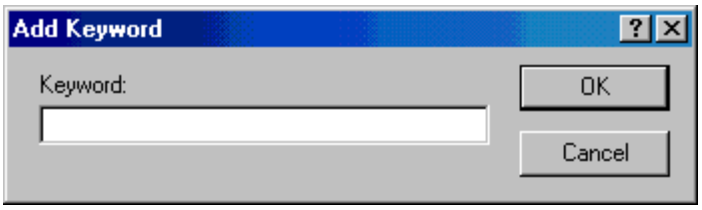
1. Select the [Setup–>Format Views...] menu option. The **Format Views** dialog will be displayed. Expand the **Source** view icon in the tree.
2. Select the file type for which you want to create a new keyword group from the tree on this dialog.
3. Click the **Add** button underneath the tree. The **Add Category** dialog will then be displayed. Enter the name of the group into the **Category Title** field. Click the **OK** button to add a keyword group. To modify the name of the group, select the keyword group and click the **Modify** button underneath the tree. **Modify Category** dialog will then be displayed. Enter the name of the group into the **Category Title** field. To remove a keyword group from the tree, select the keyword group and click the **Remove** button underneath the tree.

To create new keywords:

1. Select the [Setup–>Format Views...] menu option. The **Format Views** dialog will be displayed.
2. Select the desired keyword group to be modified and click the **Keywords** tab.



3. Click the **Add** button to add a keyword. Then the **Add Keyword** dialog will be launched. Specify a keyword in the **Keyword** field and click the **OK** button to close the dialog. To remove a keyword from the Keywords list, select the keyword and click the **Remove** button.



4. Click the **OK** button.

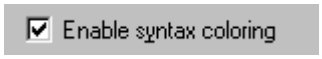
Note:

On the **Keyword** field of the **Add Keyword** dialog box, specify a keyword which consists of only alphanumerics, an underscore, and the # character.

4.12.4 Enabling/disabling syntax coloring

To enable/disable Syntax Coloring:

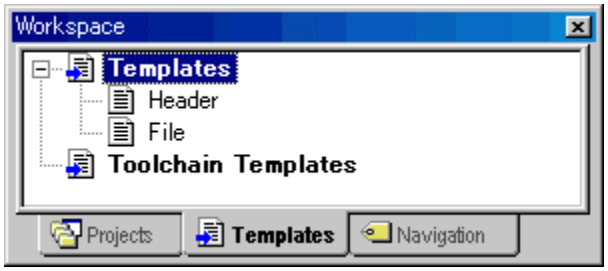
1. Select the [Setup–>Options...] menu option. The [Options] dialog will be displayed. Select the [Editor] tab.
2. Set the [Enable Syntax Coloring] checkbox as necessary and then click the [OK] button.



4.13 Templates

When developing software it is often necessary to enter the same text repeatedly, for instance, when typing a function definition, for loop or a comment block for a function. The HEW editor allows you to specify a block of text (or template) which can be inserted into the currently active **Editor** window. Thus, once a template has been defined, it can be automatically inserted without the need to re-enter it manually.

Figure below shows a list of templates, which is located on the [Templates] tab of the Workspace window.

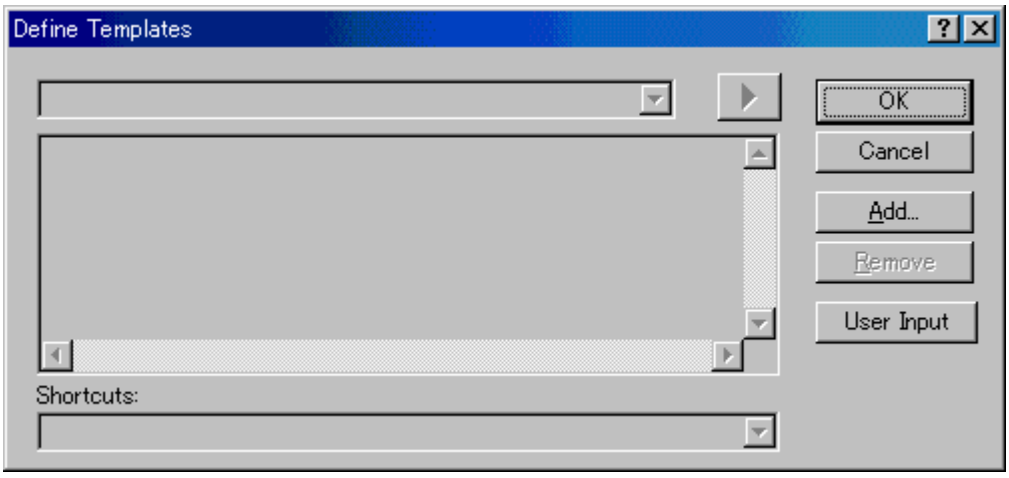


Any new templates, which have been added to the HEW, are displayed under the templates folder. The toolchain templates folder is for templates, which are read only and have been provided for use in the HEW system by the current toolchain. Templates in this view can be dragged for insertion into an editor file. It is also possible to drag an area of text from the editor into the templates folder for quick template creation. Right clicking on this view displays a pop-up so that you can quickly add a new template, remove the current selection and edit the current selection.

4.13.1 Defining a Template

To define a template:

1. Select one of the following operations to display the **Define Templates** dialog:
   - Click the Define Template toolbar button , **OR**
   - Select the [Edit–>Templates–>Define Templates...] menu option, **OR**
   - Right-click and select the [Templates–>Define Templates...] menu option from the pop-up menu.



2. Click the **Add...** button. A dialog is displayed, which asks you to enter your chosen template name. This name must be unique, otherwise a duplicated template name message will be displayed and the template will not be added.

3. If you want to modify an existing template use the **Template Name** drop-down menu to select which template you want to modify.
4. Enter the desired text into the **Template Text** area. You can copy text from another **Editor** window and then paste it into this dialog using CTRL+V.
5. To insert special information when the template is inserted, enter the keywords from the table below.
6. Enter '`$(^)`' to specify where the insertion cursor is to be placed after the template has been inserted. If this is not specified then the insertion cursor will be placed after the last character in the template, as in a normal paste operation.
7. There are 10 shortcut keys reserved for templates. If you want to designate one of these select the key in the drop list at the bottom of the edit template dialog. These range from ALT+0 to ALT+9.

Menu Entry	Placeholder	Replaced With
Time	$(TIME)	Current time
Date as DMY	$(DATE_DMY)	Current date in dd/mm/yy form
Date as MDY	$(DATE_MDY)	Current date in mm/dd/yy form
Date as YMD	$(DATE_YMD)	Current date in yy/mm/dd form
Date as Text	$(DATE_TEXT)	Current date in text form
Line	$(LINE)	First line number of template insertion
User	$(USER)	Current Windows® user
File	$(FULLFILE)	Name of the file
Project Name	$(PROJNAME)	Current project name
Workspace Name	$(WORKSPNAME)	Workspace name
Cursor Position	$(^)	Insertion cursor – positions the cursor in this position after template has been inserted

4.13.2 Deleting a Template

To delete a template:

1. Select one of the following operations to display the **Define Templates** dialog:
   - Click on the Define Template toolbar button  , **OR**
   - Select the [Edit→Templates→Define Templates...] menu option, **OR**
   - Right-click and select the [Templates→Define Templates...] menu option from the pop-up menu.
2. Use the **Template Name** drop-down list to select the name of the template you wish to remove and then click the **Remove** button.
3. Clicking the **OK** button saves the changes and closes the dialog.

4.13.3 Inserting a Template

To insert a template:

1. Select one of the following operations to display the **Insert Template** dialog:
   - Click the Insert Template toolbar button  , **OR**
   - Select the [Edit→Templates→Insert Template] menu option, **OR**
   - Right-click and select the [Templates→Insert Templates] menu option from the pop-up menu.

2. Use the **Template Name** drop-down list to select the name of the template to be inserted, and then click the **OK** button. The dialog is closed and the chosen template is added to the current **Editor** window.

Alternatively, you can press ALT along with the number of the template to be inserted (e.g. ALT+4 to insert template 4). You can define these shortcuts on the templates dialog. A drop-down list is available.

4.14 Brace Matching

Complicated source code can often become unwieldy, especially when blocks of C /C++ code are deeply nested within each other, or when complex logic statements are expressed within a large '`if`' clause. To help in such situations, the High-performance Embedded Workshop editor provides a **Brace Matching** feature, which highlights text between braces of type { }, ( ) and [ ].

To find a matching brace:

1. Either highlight the open brace to match from or place the cursor before it.
2. Select one of the following operations:
   - Click the Match Braces toolbar button  , **OR**
   - Press SHIFT+CTRL+M, **OR**
   - Select the [Edit→Match Braces] menu option, **OR**
   - Right-click on the open brace to match from and select the [Match Braces] menu option from the pop-up menu.

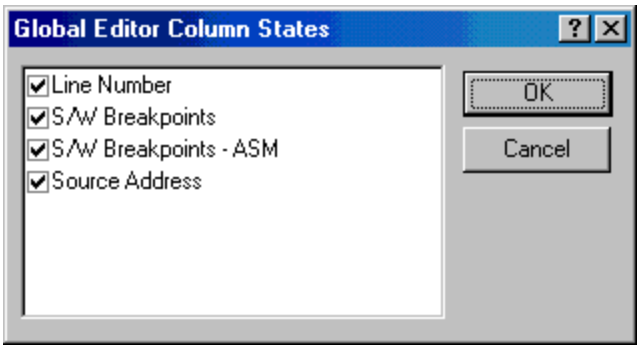
To check the structure of an entire file, place the cursor at its start and then repeatedly invoke the match brace operation. The editor will successively highlight each pair of braces in turn until there are no more left to match.

4.15 Editor Column management

The editor in HEW has the ability to manage columns (apart from the main editor column). These can be added and used by any component in the HEW system. Examples of this functionality might include a hardware breakpoint column added by the target, or possibly an address information column added by the debugger. The global column states feature is also accessible from the main edit menu.

To switch off a column in all source files:

1. Right-click in the Editor window.
2. Click the [Define Column Format…] menu item. The [Global Editor Column States] dialog will be displayed.
3. If the column's checkbox is checked then the column is enabled; if the column's checkbox is gray then this means that the column is enabled in some files, and disabled in others.
4. Click the [OK] button for the new column settings to take effect.



To switch off a column in one source file:

1. Right-click in the Editor window of the file which you wish to remove a column from, and the editor pop-up is displayed.
2. Click the [Columns] menu item and a cascaded menu item appears. Each column is displayed in this pop-up menu. If the column is enabled it will have a tick next to its name. Clicking on the entry will toggle whether the column is displayed or not.

4.16 Tooltip Watch

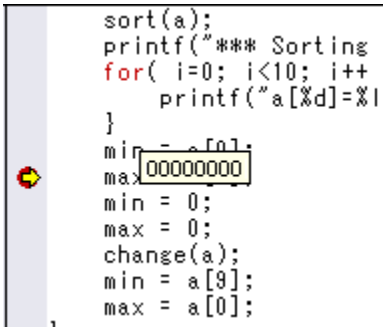
The quickest way to look at a variable in your program is to use the **Tooltip Watch** feature.

To use Tooltip Watch:

1. Select [Setup->Options…] menu option.
2. Select the [Editor] tab. The [Options] dialog box is displayed.
3. Check the [Enable tooltip watch] check box.
4. Click [OK].

To view a tooltip watch on the Editor window:

1. Open the **Source** window showing the variable that you want to examine.
2. Rest the mouse cursor over the variable name that you want to examine. A tooltip will appear near the variable containing basic watch information for that variable.



4.17 Smart edit capability in the HEW editor

Another feature of the High-performance Embedded Workshop is its smart edit facility. This is enabled by default for all C++ source files. This feature allows the HEW editor to access C++ navigation information and provide auto-completion help when using C++ classes and member functions.

To view the Smart edit status:

1. Click on the [Setup->Options...] menu item.
2. Select the Editor tab of the tools options dialog.
3. The Enable Smart-edit for C++ files should be checked.
4. Click OK.

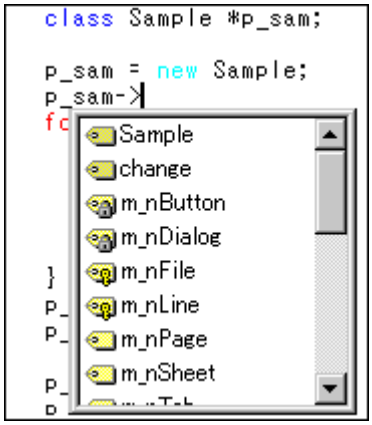
With this option switched on if you are working on a C++ file the smart-edit capability should be enabled.

Note:

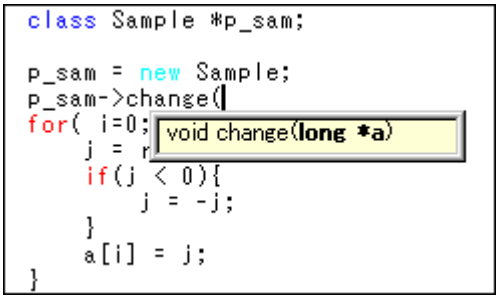
If the C++ Class navigation category has been switched off the Smart-edit capability of HEW will not be active.

During normal usage the following editor operations will make the smart edit facilities visible.

- If you are using an object and are trying to access the members using the '.' or '->'. If you do this a pop-up will be displayed which may help you select the correct member more efficiently than typing. Whilst typing the pop-up will keep track of the keys you have pressed to help your selection. If you press return then the currently selected member will be added. This pop-up is also used when using the '::' method and it is displayed in figure below.



- If you are trying to use a C++ function then the dialog in Figure below is displayed when the first open bracket is entered. This dialog allows you to see what functions are available for the current object. Selecting the function automatically enters the remaining parameters for you.



4.18 Evaluate an Expression

Launches the [Evaluate] dialog box allowing the user to enter a numeric expression, e.g. "205*2", and display the result in all currently supported radices.

To evaluate an expressions:

1. Select the [Edit->Evaluate...] menu option.
2. Launches the [Evaluate] dialog box.
3. Enter the expression that you wish to evaluate and click the [Evaluate] button.

Provides a calculator function, evaluating simple and complex expressions, with parentheses and symbols. All operators have the same precedence but parentheses may be used to change the order of evaluation. The operators have the same meaning as in C/C++. Expressions can also be used in any command where a number is required. The result is displayed in all supported radix types.

Valid operators:

Addition (+)	Subtraction (-)	Multiplication (*)	Division (/)
Logical AND (&&) ***1**	Logical OR (||) ***1**	Logical NOT (!) ***1**	Equal to (==)
Bitwise AND (&)	Bitwise OR (|)	Bitwise NOT (~)	Unequal to (!=)
Left arithmetic shift (<<)	Right arithmetic shift (>>)	Less than (<)	Greater than (>)
Modulo (%) ***1**	Bitwise exclusive OR (^)	Less than or equal to (<=)	Greater than or equal to (>=)

***1 :** Support for this function depends on the debugging platform.

Register names:

- **SH/H8/R8C (E7/E8) family**

  Register names may be used, but must always be prefixed by the '#' character.

  Enter a numeric expression, e.g. "(#pc+205)*2", and display the result in all currently supported radices.

- **M32C/M32R/M16C/R8C (excluding E7/E8) family**

  Register names may be used, but must always be prefixed by the '%' character.

  Enter a numeric expression, e.g. "(%pc+205)*2", and display the result in all currently supported radices.

Character Constants:

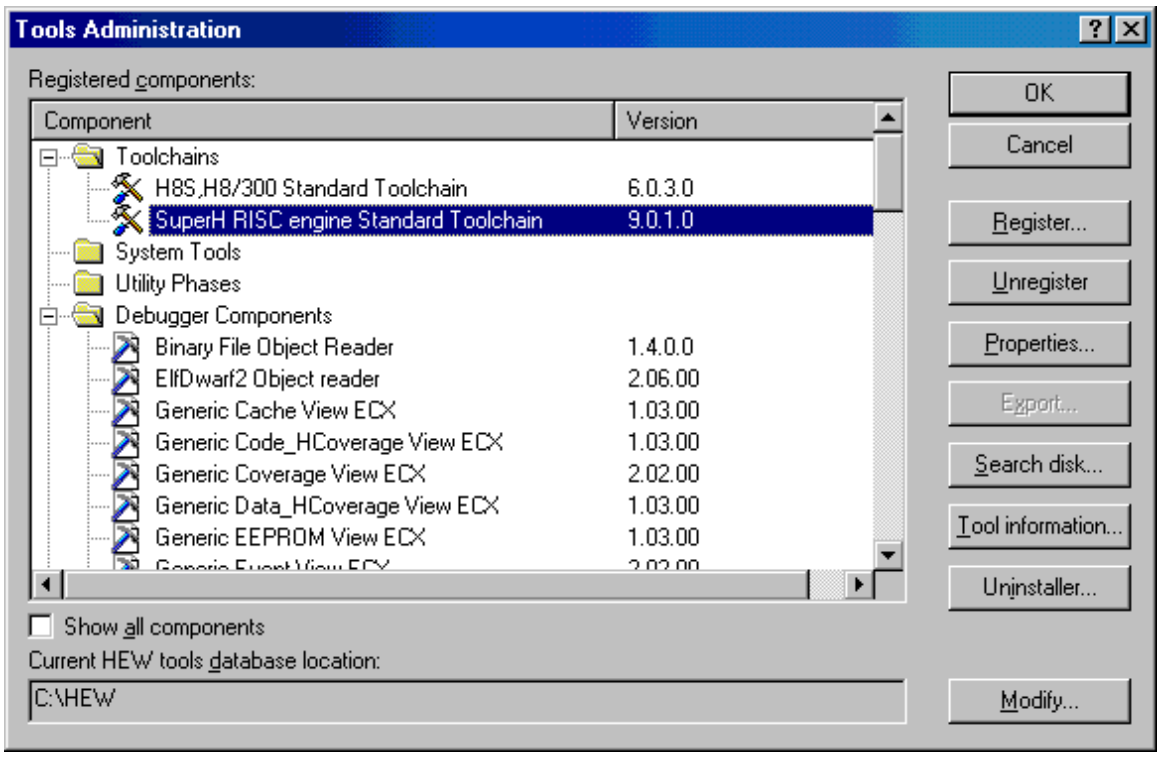
Characters enclosed in single quote marks (') may be use as character constants. For example, 'A', etc. These character constants are converted to ASCII code and used as 1-byte immediate values.

Character String Literals

Character strings enclosed in double quote marks (") may be use as character string literals. Examples are "abc", etc.

5. Tools Administration

You can control the components that are used by the High-performance Embedded Workshop via the **Tools Administration** dialog, which is invoked by selecting the [Tools→Administration...] menu option. The **Tools Administration** dialog is only accessible when no workspace is open.



There are five standard types of component:

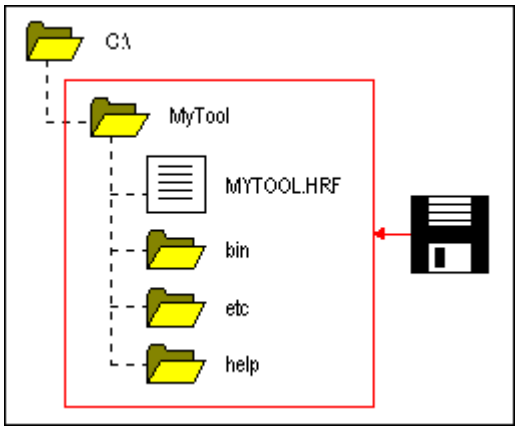
Toolchain	A Toolchain is a set of build phases (e.g. compiler, assembler, linker and librarian). These components provide the build capability.
System Tool	A System Tool is an application (`.EXE`) that can be launched from the **Tools** menu. These are often provided as extra applications, which support the toolchain (e.g. an external debugger like the Hitachi Debugging Interface or an interactive graphical librarian).
Utility Phase	A Utility Phase is a 'ready made' build phase, which supports some specific build functionality (e.g. analyze complexity of source code, count lines of source code, etc.). These components provide added functionality to the build that is not toolchain-specific.
Debugger Component	A Debugger Component is a component that supports some specific debugger functionality (e.g. CPU DLL, Target platform, Object reader, etc.).
Extension Component	An Extension Component is a component that provides key functionality in a certain area of the HEW system. These components cannot be unregistered when installed (e.g. The HEW builder, debugger and flash support).

5.1 Tool Locations

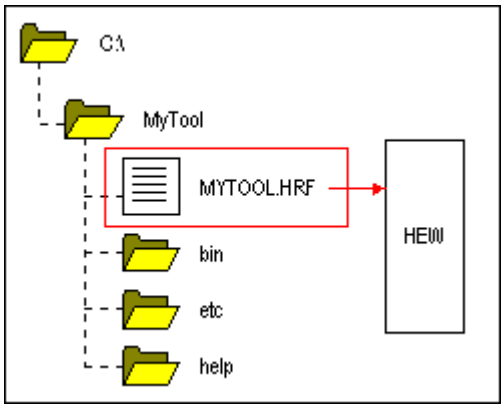
The HEW maintains the locations of HEW compatible components automatically as each new tool is installed. After installation, the HEW stores information about the component (including its location). This is referred to as **Registration**. Although initial registration is automatic, during the course of development or if you want to manage the tools being used in your projects more effectively, you may need to register components yourself.

5.2 HEW Registration Files (*.HRF)

When a HEW compatible component (i.e. toolchain, system tool or utility phase) is installed, part of its installation will include a file with the extension `.HRF`. This file, called a **HEW Registration File**, describes the component to the HEW (see the figure below).



The process of registration refers to loading a component's `.HRF` file into the **Tools Administration** dialog (see the figure below).



In order to use a component with HEW it must first be registered (see the Registering a component topic for further information). The **Tools Administration** dialog shows all currently registered components. To access it, ensure no workspaces are open and then select the [Tools→Administration] menu option. If you attempt to access the **Tools Administration** dialog when a workspace is open, the **Tools Administration** dialog is opened but cannot be modified. When HEW is installed, any new tools are automatically registered. Day to day usage of the HEW though, may mean you need to know more about the tools registration process.

HEW stores tool information in a tool database file, which is stored in the root of the tool installation directory. By default this is set to the HEW application directory, however if you are working in a network environment this directory may be set to another location. It is possible to change the tool directory location and this causes a re-scan of the tools that are registered in HEW.

To change the tools directory location:

1. Select the [Tools→Administration...] menu option.
2. Click the **Modify...** button, which is next to the **Current HEW Tools Database Location** field.
3. Browse to the root directory of the new tool location and click the **OK** button.
4. This will switch the directory and change the tool location to the new directory. It will be necessary to scan for any new tools that may be in this location. This can be achieved by using the **Search Disk** or **Register Tool** functionality.

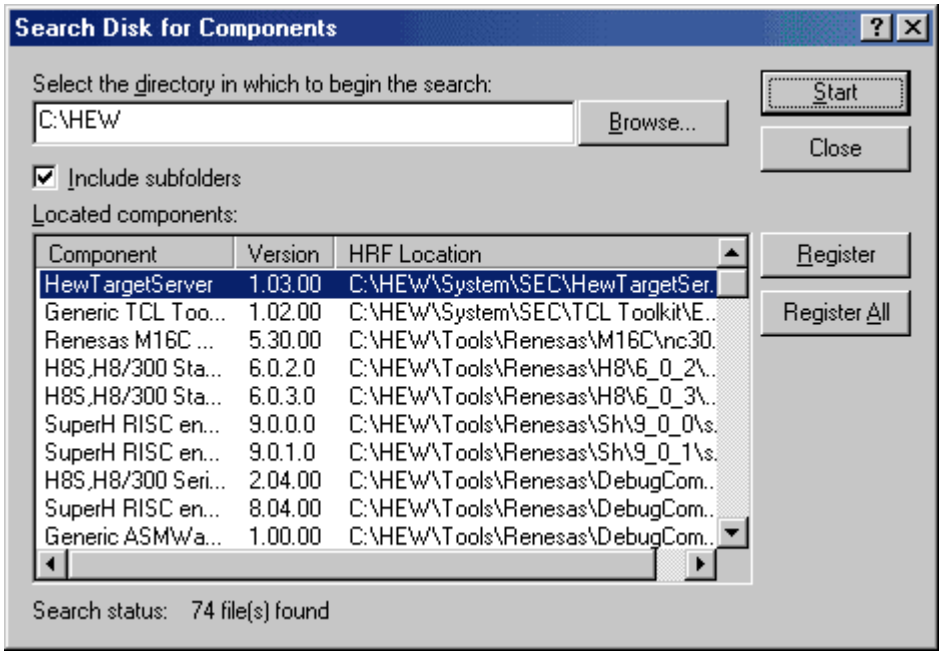
5.3 Registering a Component

The HEW will automatically attempt to register any new components installed since the last time it was invoked. However, in some circumstances you may need to register components yourself.

In some cases it is useful to search a drive for HEW-compatible components. This is especially useful if the HEW installation was deleted or corrupted, as it can recreate your tool information instantly.

To search for components and register them:

1. Click the **Search Disk** button on the **Tools Administration** dialog. The **Search Disk for Components** dialog will be displayed.
2. Enter the directory in which you would like to search into the top field, or browse to it graphically by clicking the **Browse** button.
3. Check the **Include Subfolders** checkbox if you would like to search the directory specified and all directories below it.
4. Click the **Start** button to begin the search. During the search, the **Start** button will change to a **Stop** button. Click the **Stop** button to halt the search at any time.
5. The results of the search are shown in the **Located Components** list. Select a component and click the **Register** button to register an individual component, or click the **Register All** button to register all located components.
6. Click the **Close** button to exit the dialog.



To register a single component:

1. Click the **Register** button on the **Tools Administration** dialog.
2. Browse to the component's `.HRF` file and click the **OK** button to register that component.

Note:

The HEW Registration File (*.HRF) is located in the root directory of a component's installation.

5.4 Unregistering a Component

The components that are registered with the HEW affect the way it behaves. For example, every compatible system tool that is registered will be added to the **Tools** menu when a new project is created. Sometimes this may not be desirable. If this is not required, open the **Tools Administration** dialog, select the undesired component from the **Registered Components** list and click the **Unregister** button. A dialog will be invoked, which asks you to confirm this action. Click **Yes** to unregister the component.

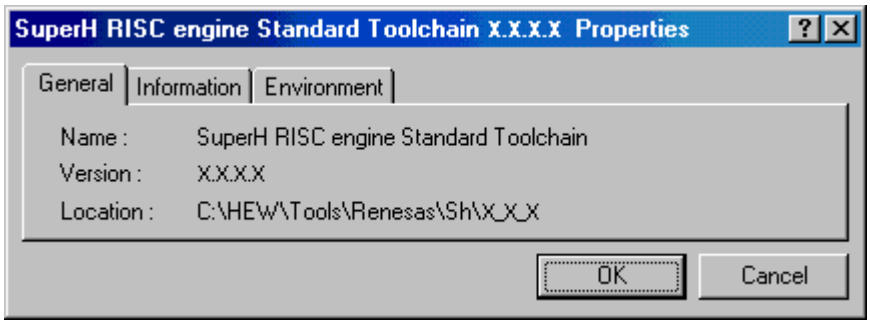
Note:

Unregistering a component does not remove its installation from the hard disk. It simply removes the information that the HEW was storing about that component (i.e. it 'disconnects' it from the HEW). The action can be easily reversed at any time by registering the tool manually (see section 5.3, Registering a Component). If you want to remove a component from the hard disk (i.e. uninstall a component) then refer to section 5.6, Uninstalling a component.

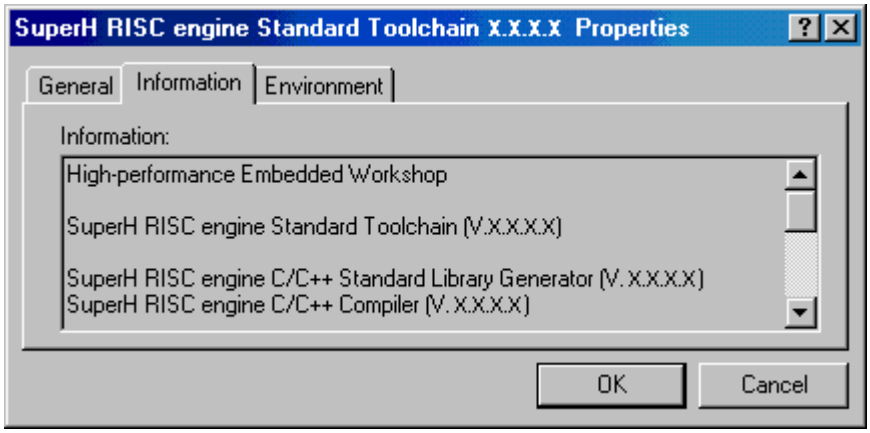
5.5 Viewing and Editing Component Properties

To view information regarding a component, select it from the **Registered Components** list on the **Tools Administration** dialog and click the [Properties...] button. The **Properties** dialog will be displayed.

The **General** tab displays the name, version and location of the selected component.

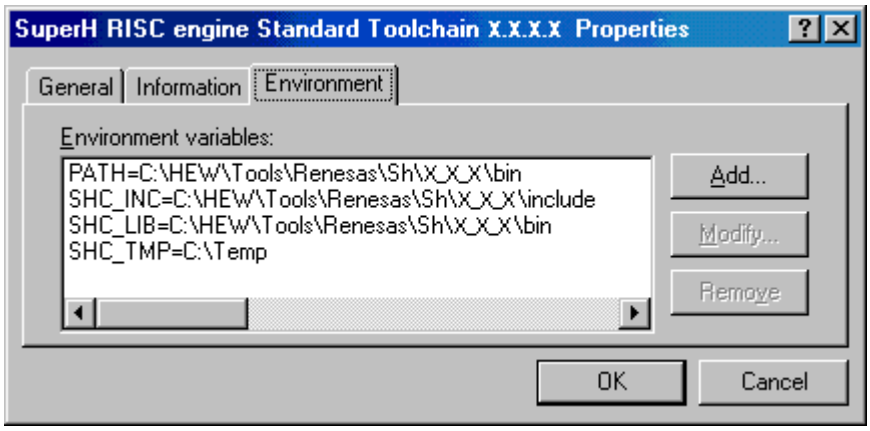


Select the **Information** tab to view any information about the component. This may include copyright information, enhancements and so on.



If there is an issue with the component and it is working incorrectly additional information is displayed here.

Select the **Environment** tab, if it exists, to view and edit a component's environment settings. This tab is most commonly used to modify the environment of a toolchain.

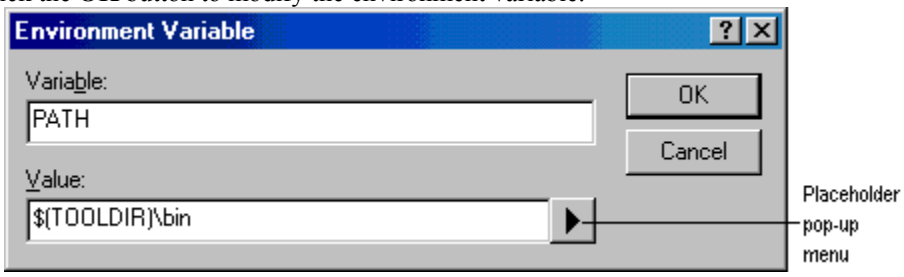


To add a new environment variable:

1. Click the [Add...] button. The **Environment Variable** dialog will be invoked.
2. Enter the variable name into the **Variable** field.
3. Enter the variable's value into the **Value** field
4. Click the **OK** button to add the new variable to the **Environment** tab.

To modify an environment variable:

1. Select the variable that you want to modify from the **Environment** tab.
2. Click the [Modify...] button. The **Environment Variable** dialog will be invoked.
3. Make the required changes to the **Variable** and **Value** fields.
4. Click the **OK** button to modify the environment variable.



To remove an environment variable:

1. Select the variable that you want to remove from the **Environment** tab.
2. Click the [Remove] button.

Note:

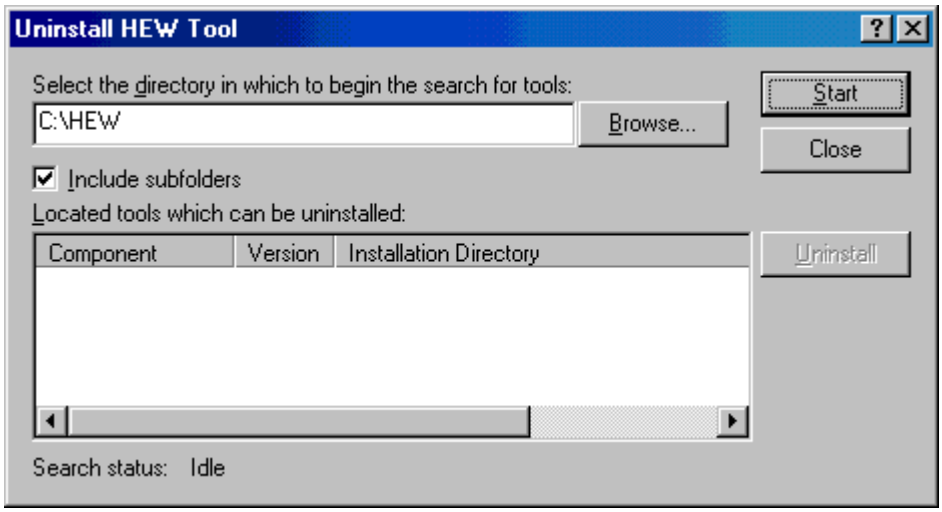
Placeholder pop-up menus are included to ensure that the environment can be specified as flexibly as possible. For further information about using placeholders, see Reference 4, Placeholders.

5.6 Uninstalling a component

The HEW provides a built-in Tools Uninstaller, which can remove unregistered components.

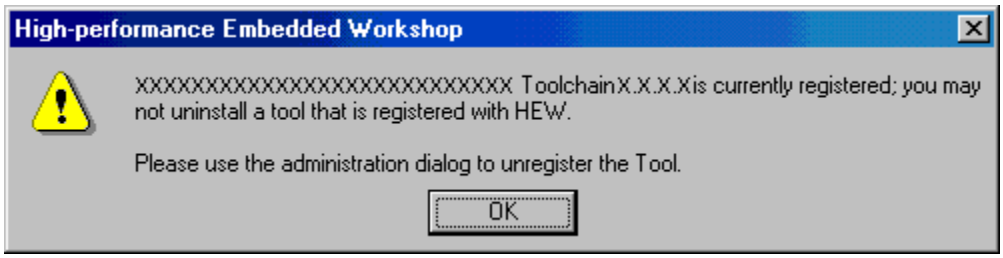
To run the Tools Uninstaller:

1. Select the [Tools–>Administration...] menu option.
2. Click the [Uninstaller...] button. The **Uninstall HEW Tool** dialog is displayed.



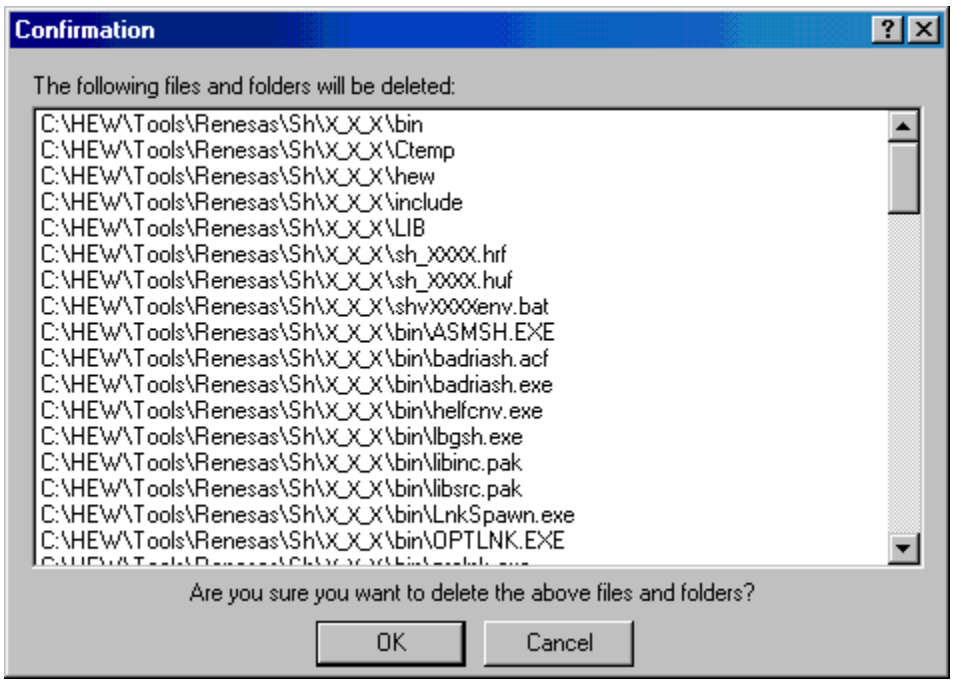
3. Enter the directory in which you would like to search into the search field, or browse to it graphically by clicking the [Browse...] button.
4. Check the **Include subfolders** checkbox if you would like to search the directory specified and all directories below it.
5. Click the [Start] button to begin the search. During the search, the [Start] button will change to a [Stop] button. Click the [Stop] button to halt the search at any time.
6. The results of the search are shown in the **Located tools which can be uninstalled** list. Select a component and click the [Uninstall] button to uninstall that component.
7. Click the [Close] button to exit the dialog.

A component may only be uninstalled if it is not currently registered with the HEW. If you attempt to uninstall a tool that is registered, you will be asked to uninstall the tool first.



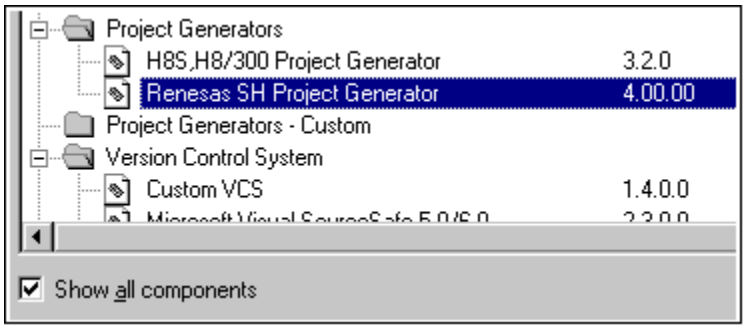
You must then return to the **Tools Administration** dialog (accessed by selecting the [Tools–>Administration...] menu option), unregister the tool and then invoke the **Tool Uninstaller** again.

If the selected tool is not registered with the HEW, the [Confirmation] dialog will be displayed when the [Uninstall] button is clicked. This dialog displays all of the files and folders that will be deleted. If you are certain that these files and folders can be deleted then click the **OK** button. To abort the uninstallation, click the **Cancel** button.



5.7 Technical support

The **Tools Administration** dialog is capable of displaying information regarding 'hidden' system components. These are part of the HEW itself, and cannot be registered/unregistered manually. If you check the **Show all components** checkbox on the **Tools Administration** dialog, extra component folders are displayed (see the figure below).



When seeking technical support, you may be asked to give details about some or all of these components. To do so, open the respective folder, select a component and click the **Properties** button. The **Properties** dialog will be invoked.

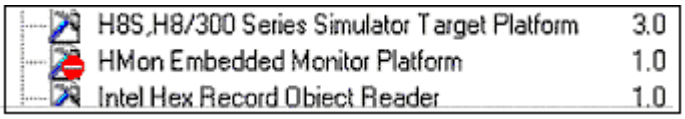
The HEW also has a feature that outputs tool information regarding the registered components to a file. This allows you to retrieve information about the entire HEW system. This information can then be sent to your technical support contact if you are experiencing problems with the HEW.

To output tool information:

1. Click the [Tools–>Administration...] menu item.
2. Click the [Tool information...] button. A [Save tool information file] dialog is displayed.
3. Choose the location of the output file and click the **OK** button.
4. A file is created in the chosen location with the current registered tool setup of the HEW.

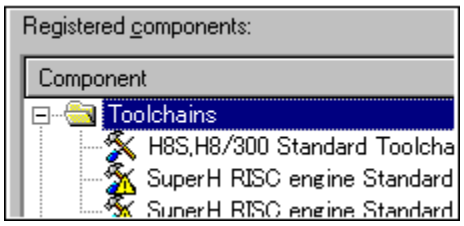
If any of the components have problems these can be seen in the tools administration dialog. If the icon has an additional icon this explains the problem. There are two additional icons that can be displayed.

If a component is found but cannot be used due to it being an old version or another dependent component is not available then the icon in figure below is used to show this.



[Component not found] icon

If the component is not located where the registration file says it is then the icon in figure below is used to show this.



[Incompatible component found] icon

Note:

If the tool has one of these errors then it is possible to get more information by the following method:

To get tool error feedback:

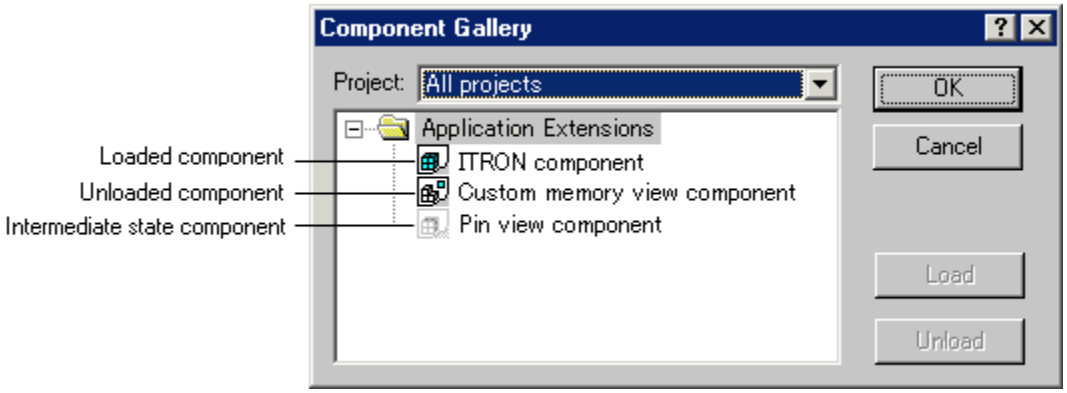
1. Click the [Tools->Administration...] menu item.
2. Select the tool in the list that has an issue.
3. Click [Properties…].
4. Select the information tab and scroll the edit field to the bottom.
5. The reason for the problem will be displayed in this area.

5.8 Using On-Demand components

The HEW version 3.0 onwards has the concept of on-demand components. These components are not automatically loaded by the application or the debugger component. These components can be loaded by the user or as part of the project generation process.

To load or unload an on-demand component manually:

1. Click the [Project->Components…] menu item.
2. The component gallery dialog is displayed.
3. Select the component you wish to load. Click the load button. The components image should change to the loaded state.
4. If you wish to unload a component. Select the component. Click the unload button. The components image should change to the unloaded state.
5. Click OK to verify the changes.



Note:

Each project in your workspace can have different components loaded and unloaded. If you have multiple projects you can use the “Multiple projects…” and “All projects” items to change a components load status over more than one project. If you select a combination which means the component is loaded in one project and not anther then the intermediate state icon is displayed.

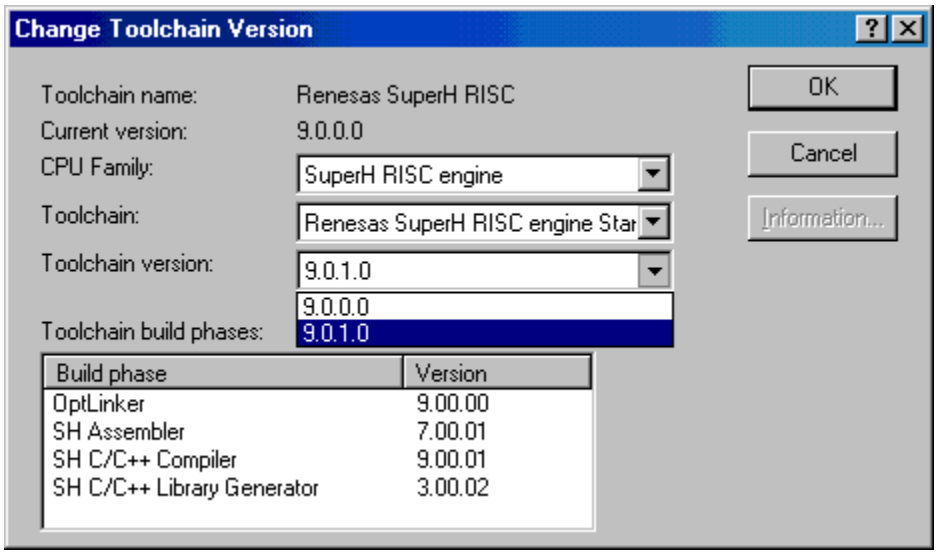
5.9 Custom project types

The [Project→Create Project Type] menu item in HEW and allows you to create a template for your project. This menu item takes the settings of the current project and then creates a project type for you. The user can specify the name of the new type and the style of the Project Generation Wizard. Once created, these project types appear in the **Tools Administration** dialog and are initially hidden in the System Components part of the **Tools Administration** tree.

To export one of the Custom Project Generators, select the **Export** button on the **Tools Administration** dialog. The export functionality packages the Custom Project Generator into a binary file, which includes an executable. This can be given to another user who then runs the executable. This installs the Project Generator into the correct location on the target user's machine.

Note:

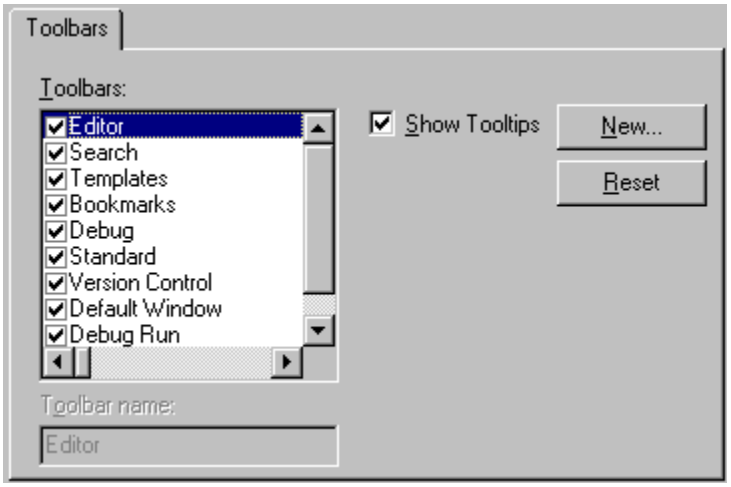
A project template can be created by selecting [Project–>Create Project Type...] based on the project in use. This template includes the version information of the toolchain. When a project is created by using this template after the toolchain version has been updated in your HEW system, check that the toolchain version of the created project matches the using environment. When the registered toolchain can be updated, the toolchain version can be changed in the dialog box that is displayed by selecting [Tools–>Change Toolchain Version...].



6. Customizing the Environment

6.1 Customizing the toolbars

The High-performance Embedded Workshop provides standard toolbars as detailed in the Toolbars topic. In addition to these, you may also construct your own toolbars.



To create a new toolbar:

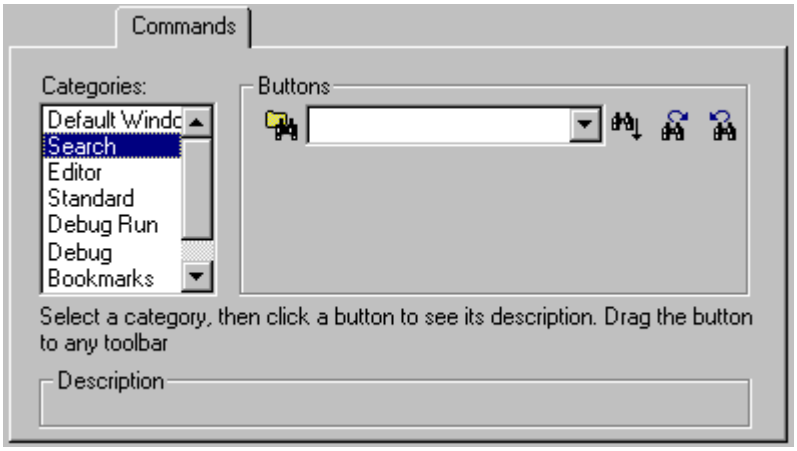
1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed.
2. Click the **New** button. The **New Toolbar** dialog will be displayed.
3. Enter the name of the new toolbar into the **Toolbar Name** field.
4. Click the **OK** button to create the new toolbar.

Note:

When a new toolbar is created it will appear undocked (i.e. 'floating') and empty.

To add buttons to a toolbar:

1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed. Select the **Commands** tab.
2. Browse the available buttons by selecting the button categories from the **Categories** list. Select a button from the **Buttons** area to display information on its operation.
3. Click and drag a button from the dialog onto the toolbar.



To remove buttons from a toolbar:

1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed. Select the **Commands** tab.
2. Click and drag a button from the toolbar onto the **Buttons** area.

To modify the name of a user-defined toolbar:

1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed.
2. In the **Toolbars** list, select the user-defined toolbar and whose name you want to modify.
3. Modify the name of the toolbar in the **Toolbar Name** field.
4. Click the **OK** button to save the toolbar's new name.

To remove a user-defined toolbar:

1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed.
2. Select the user-defined toolbar from the **Toolbars** list and the **Reset** button will change to a **Delete** button. Click the **Delete** button.

To reset a standard toolbar back to its original state:

1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed.
2. Select the standard toolbar from the **Toolbars** list and then click the **Reset** button.

To show or hide toolbar tooltips:

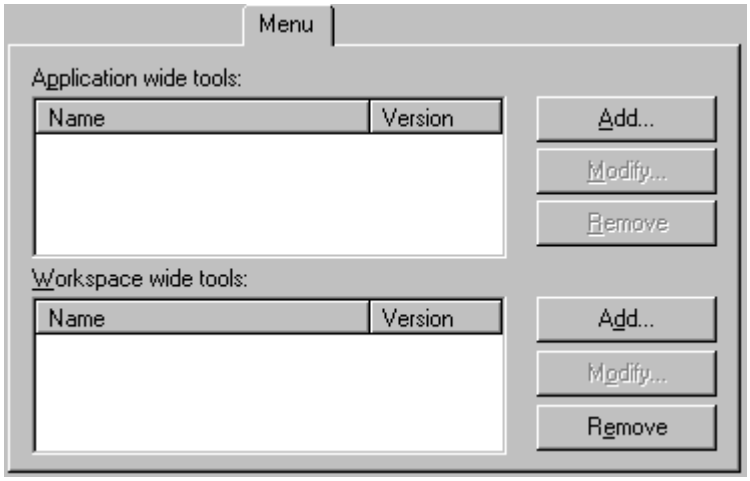
1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed.
2. Set the **Show Tooltips** checkbox as desired.

6.2 Customizing the tools menu

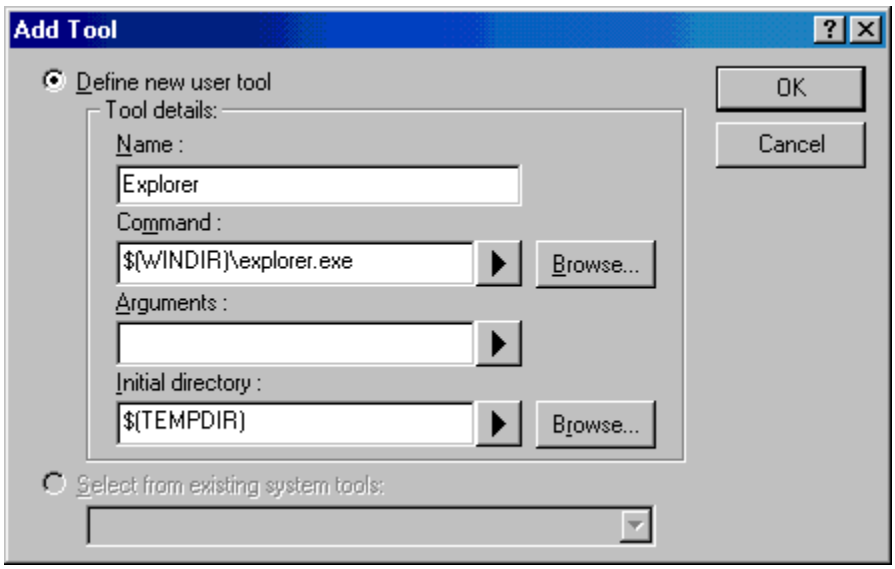
The **Tools** menu can be customized to include your own menu options.

To add a new menu option:

1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed. Select the **Menu** tab. The first thing you need to decide is whether you are adding a global application-wide tool (which will be available to all of your workspaces), or whether you wish to add a workspace-wide tool (which is only valid for the current workspace). Once you have chosen, choose the relevant section of the dialog.



2. Click the **Add** button. The **Add Tool** dialog will be invoked. If you would like to add an existing system tool to the menu then select the **Select from existing tools** radio button, choose the tool from the drop-down list and then click the **OK** button. Alternatively, if you would like to add a tool of your own then follow the remaining steps.
3. Enter the name of the tool into the **Name** field.
4. Enter the command, excluding arguments, into the **Command** field.
5. Enter any arguments that you would like to pass to the command into the **Arguments** field.
6. Enter the initial directory, in which you would like the tool to run, into the **Initial Directory** field.
7. Click the **OK** button to add the menu option to the **Setup** menu.

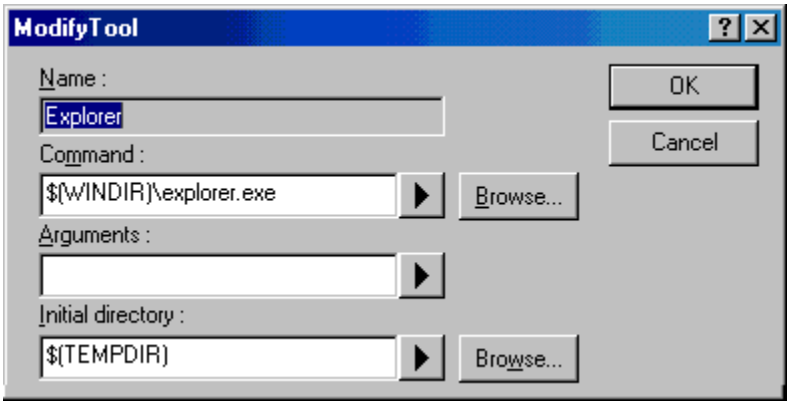


Note:

New menu options are added to the bottom of the list (i.e. bottom of the **Tools** menu) by default. However, the order of menu options in the **Setup** menu can be modified.

To modify a menu option:

1. Select the [Setup–>Customize...] menu option. The [Customize] dialog will be displayed. Select the **Menu** tab.
2. Select the menu option that you would like to modify and then click the **Modify** button.
3. Make the desired changes on the **Modify Tool** dialog and then click the **OK** button.



To remove a menu option:

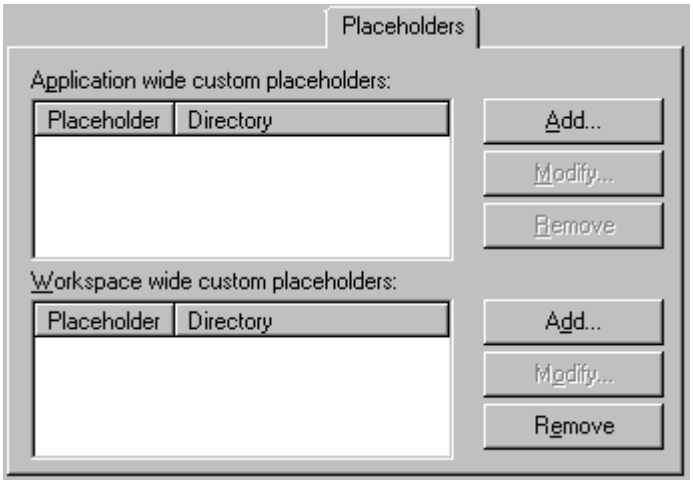
1. Select the [Setup–>Customize...] menu option. The [Customize] dialog will be displayed. Select the **Menu** tab.
2. Select the menu option that you would like to remove and then click the **Remove** button.

6.3 Using Custom Placeholders

Throughout the High-performance Embedded Workshop the user can use a number of pre-defined placeholders for directory definitions. For example, the user can use the "$(PROJDIR)" variable to signify the current HEW project directory. This makes it much easier to relocate projects and keep all of the paths correct.

The High-performance Embedded Workshop also has the ability to define custom placeholders. This means you can enter your own custom placeholder definition and decide upon its directory value. Once defined this placeholder becomes available throughout the rest of the HEW system.

The placeholders can be defined on an application-wide level so the placeholders are available to all workspaces and projects that use the HEW. The other method of defining the placeholders is using the workspace-wide custom placeholders. This means the placeholders can only be used in the current workspace. This list is only available when you have a workspace open.

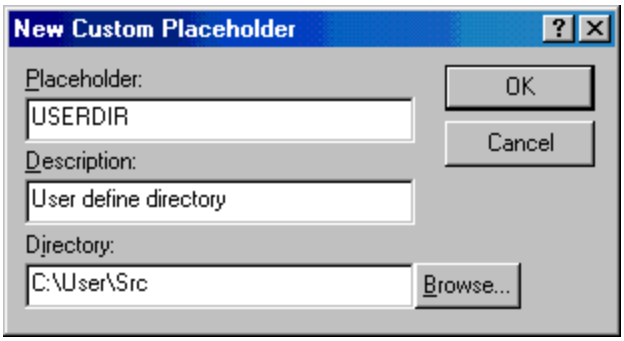


To add a custom placeholder:

1. Choose the [Setup–>Customize...]. The **Tools Customize** dialog will be displayed. Select the **Placeholders** tab.
2. Choose whether you need to use an application-wide or workspace-wide placeholder. Click the **Add** button which is adjacent to the list you require.
3. The **Add New Placeholder** dialog is displayed.
4. In the fields provided choose a suitable name for the placeholder and a description of what the placeholder means.
5. Choose a directory, which relates to this placeholder. It is possible to use placeholders that are already defined in this field, such as $(PROJDIR).

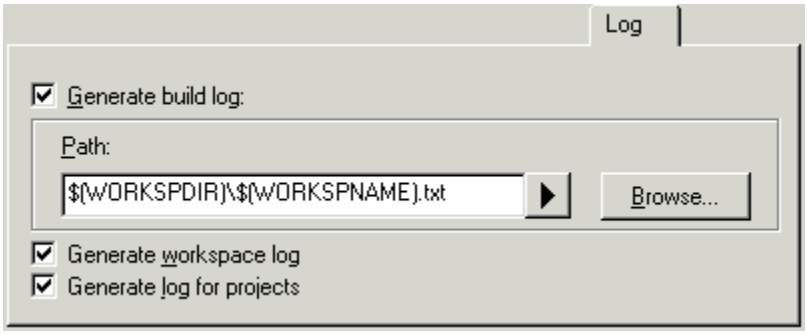
Note:

In HEW2.1 or later version, the user-defined directory can be set as the custom placeholder, which can be used for setting the toolchain option. When the directory path is specified, specify an absolute path in 'Directory', as shown in figure below.



6.4 Using the workspace and project log facilities

The HEW has workspace and project logging facilities integrated into the application. These facilities can be switched on via the log tab on the Tools customize dialog. This option is especially useful when the network database is in operation. This is because user names and changes are logged to this file.



When the [Generate workspace log] is clicked any workspace changes will be logged to a file with the same name as the workspace with a ".log" extension. This file will be located in the same directory as the workspace file.

When the [Generate log for projects] is clicked any projects in the current workspace that have changes made to them will be logged to a file with the same name as the project with a ".log" extension. This file will be located in the same directory as the project file.

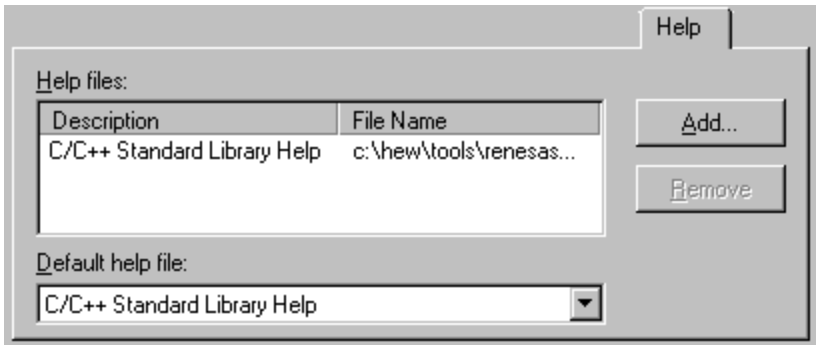
The log file is updated when the workspace is saved.

6.5 Configuring the help system

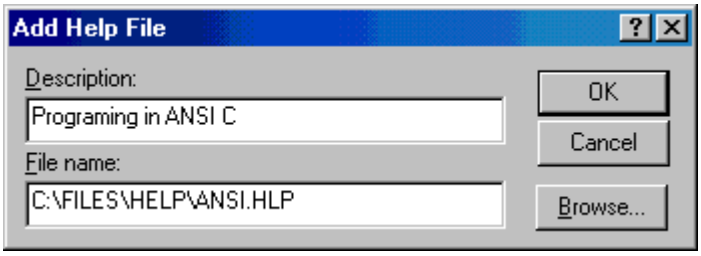
The High-performance Embedded Workshop provides context sensitive help within the **Editor** window. In other words, if you select some text in the **Editor** window and then press F1, the High-performance Embedded Workshop will attempt to locate help on that selected item. The help files that will be searched are listed in the **Help** tab of the **Setup Customize** dialog.

To add a new help file:

1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed. Select the **Help** tab.



2. Click the **Add** button. The **Add Help File** dialog will be displayed.
3. Enter a description of the help file into the **Description** field.
4. Enter the full path of the help file into the **File name** field (or browse to it graphically by clicking the **Browse** button).
5. Click the **OK** button to add the new help file to the list.



To make a help file the default choice, select it from the **Default Help File** drop-down list or set it to **None** if you would like to be prompted for a help file whenever F1 is pressed.

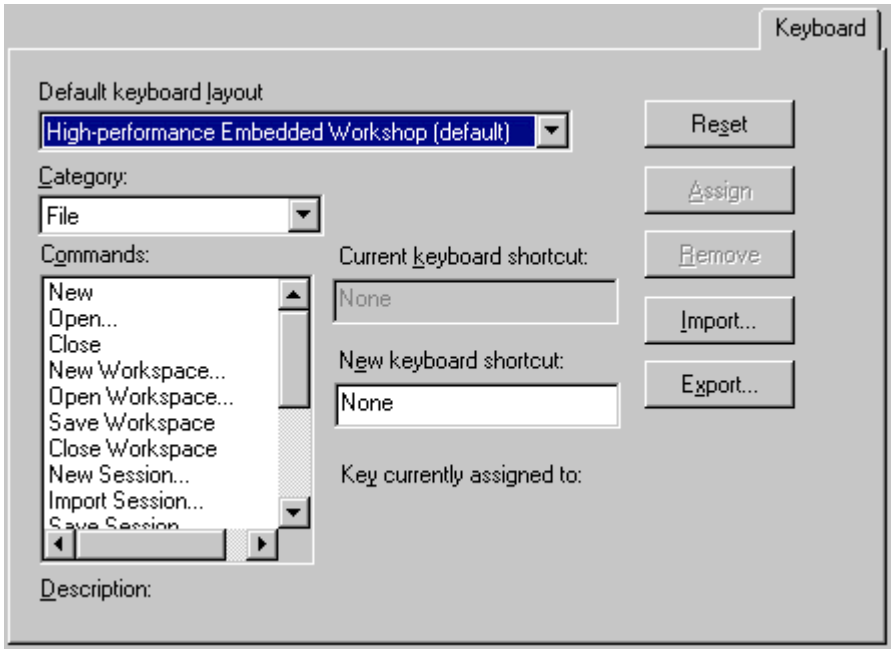
To remove a help file:

1. Select the [Setup–>Customize...] menu option. The **Setup Customize** dialog will be displayed. Select the **Help** tab.
2. Select the help file to be removed and then click **Remove**.
3. Click the **OK** button to confirm the new help file settings.

6.6 Keyboard shortcut customization

The HEW allows the keyboard shortcuts to be customized to your own preferences. This means that major operations can be configured to different keys especially useful if you are migrating from a different tool.

To reach the keyboard shortcut customization dialog click the [Setup–>Customize...] menu item. Then when [Customize] dialog is invoked click the [Keyboard] tab.



This dialog allows instant selection of either the default HEW keyboard shortcuts or the PD debugger shortcuts. To change the entire keyboard layout select an item in the [Default keyboard layout] list. By default it uses the [High-performance Embedded Workshop] settings.

A number of operations are possible on this dialog:

To add a new keyboard shortcut:

1. Select the main menu category of the command you wish to modify. It is only possible to modify the commands that have a menu. Only some cases are special that allow modification, these are named global.
2. Select the command you wish to modify or add a new keyboard shortcut for in the [Commands] list.
3. The current shortcut is displayed in the [Current keyboard shortcut] field.
4. Enter the new shortcut in the [New keyboard shortcut] field. Various combinations of buttons can be used here. For example "CTRL+J" or "SHIFT+CTRL+O", etc. If the chosen shortcut is already in use it is displayed below the New keyboard shortcut field.
5. If you are happy with your selection click the [Assign] button.
6. Changes are not saved until the [OK] button is clicked on the [Customize] dialog.

To remove a keyboard shortcut:

1. Select the main menu category of the command you wish to modify. It is only possible to modify the commands that have a menu. Only some cases are special that allow modification, these are named global.
2. Select the command you wish to modify or add a new keyboard shortcut for in the [Commands] list.
3. The current shortcut is displayed in the [Current keyboard shortcut] field.
4. Click the [Remove] button.
5. Changes are not saved until the OK button is clicked on the [Customize] dialog.

To reset all the keyboard shortcuts:

1. Click the [Reset] button. All shortcuts revert to the default settings for the currently selected keyboard layout.
2. Changes are not saved until the [OK] button is clicked on the Setup customize dialog.

The keyboard shortcuts dialog allows you to import and export keyboard settings to a defined file. This allows you to easily transfer settings from one machine to another.

To export keyboard shortcuts:

1. Click the [Export...] button.
2. A standard file dialog is displayed. Choose the filename to save the settings of the currently selected keyboard layout to.
3. Click [OK].

To import keyboard shortcuts:

1. Select the keyboard layout you wish to replace with your imported settings in the [Default keyboard layout] list.
2. Click the [Import...] button.
3. A standard file dialog is displayed. Choose the filename to load the keyboard layout from.
4. Click [OK].

6.7 Scope of a Control in the Setup

6.7.1 Scope of a Control in the Customize Dialog

The scope of each control in the [Customize] dialog box, which is launched via [Setup->Customize…], differs. This can be confusing so these have been listed below:

Tab	Control	Scope
Toolbar	All	Each workspace
Command	All	The whole system
Menu	Application wide tools	The whole system
	Workspace wide tools	Each workspace
Placeholders	Application wide custom placeholder	The whole system
	Workspace wide placeholder	Each workspace
Debugger	Debugger tool	Each project
	Debugger location	Default: whole system and each project basis
	Command line options	Each project
	Session file	Each project
	Download module	Each project
Log	All	Each workspace
Help	All	Each workspace
Keyboard	All	The whole system

6.7.2 Scope of a Control in the Options Dialog Box

Scope of each control of each tab of the [Options] dialog box, which is launched via [Setup->Options…], affects the whole system.

6.8 Specifying workspace options

The High-performance Embedded Workshop allows you to control several aspects of a workspace via the **Setup Options** dialog. To invoke it select the [Setup→Options...] menu option, and select the **Workspace** tab.

6.8.1 Open last workspace at start-up

When you exit the High-performance Embedded Workshop, the last workspace you had open is stored. On subsequently launching the HEW, you may want the last workspace to be opened automatically.

To open the last workspace at start-up:

1. Select the [Setup–>Options...] menu option. The **Setup Options** dialog will be displayed. Select the **Workspace** tab.
2. Set the **Open last workspace at start-up** checkbox if you would like the HEW to automatically open the last workspace when it is launched.
3. Click the **OK** button to confirm the new settings.

6.8.2 Restore files on opening a workspace

When you close a workspace, the HEW stores the names of the files that were open at that time. When you open a workspace, the HEW can restore (i.e. open) the same files so that you can continue your session in exactly the same state as when you left it. If you would like the files associated with a workspace to be opened when the workspace is opened, then set this checkbox.

To restore files on opening a workspace:

1. Select the [Setup–>Options...] menu option. The **Setup Options** dialog will be displayed. Select the **Workspace** tab.
2. Set the **Restore files on opening workspace** checkbox if you would like the files associated with a workspace to be opened when the workspace is opened.
3. Click the **OK** button to confirm the new settings.

6.8.3 Display workspace information on opening a workspace

When many workspaces are being used, it is sometimes difficult to remember exactly what is contained within each workspace. To help resolve this, the High-performance Embedded Workshop allows you to enter a textual description of each workspace. This description can be displayed whenever a workspace is opened.

To enter a workspace description:

1. Select the workspace icon from the **Projects** tab of the **Workspace** window.
2. Right-click to invoke the pop-up menu and then select the **Properties** option. The **Workspace Properties** dialog will be displayed.
3. Enter the description into the **Information** field.
4. Set the **Show workspace information on workspace open** checkbox if you want a **Workspace Properties** dialog to be launched on opening a workspace. This checkbox has the same role as the **Display workspace information dialog on opening workspace** checkbox on the **Workspace** tab of the **Tools Options** dialog.
5. Click the **OK** button to confirm the new settings, or click the **Cancel** button to discard the changes.

6.8.4 Save workspace before executing any phases

It is possible to force the High-performance Embedded Workshop into saving the current workspace before executing any build phases (i.e. build, build all or build file operations) or version control commands.

To save the workspace before executing any phases:

1. Select the [Setup–>Options...] menu option. The **Options** dialog will be displayed. Select the **Workspace** tab.
2. Set the **Save workspace before executing any tools** checkbox.
3. Click the **OK** button to confirm the new settings.

6.8.5 Prompt before saving a workspace

If you are using the **Save workspace before executing any phases** function, you may want the HEW to prompt you before saving the workspace. For further information about saving a workspace before executing phases, see section 6.8.4, Save workspace before executing any phases.

To display a prompt before saving the workspace:

1. Select the [Setup–>Options...] menu option. The **Tools Options** dialog will be displayed. Select the **Workspace** tab.
2. Set the **Prompt before saving workspace** checkbox.
3. Click the **OK** button to confirm the new settings.

6.8.6 Prompt before saving session

Checking this option will force the High-performance Embedded Workshop into displaying a prompt before the session is saved to disk.

6.8.7 Auto-backup facilities

The HEW supports the facility to backup the workspace, project and session files at regular intervals. This means that if your application or development system should fail you will not lose so much work. Any changes you have made will be saved to temporary files.

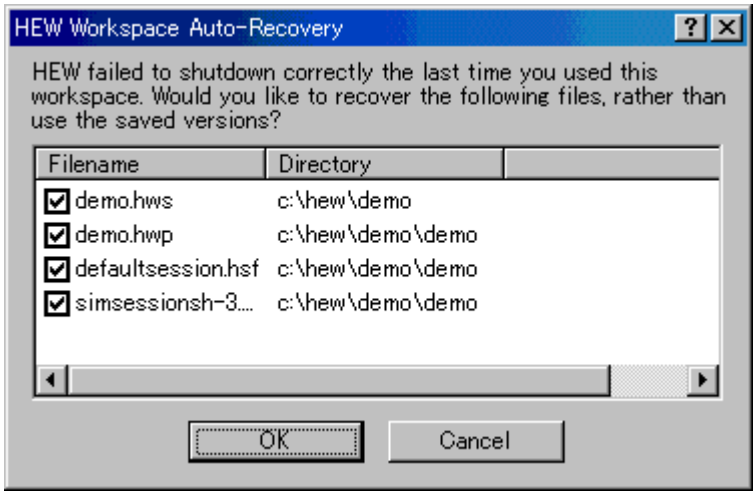
When re-opening the workspace you will be prompted and asked if you wish to auto-recover the files that were not saved during your last session.

To enable auto-file backup facilities:

1. Select the [Setup–>Options...] menu option. The **Options** dialog will be displayed. Select the **Workspace** tab. This dialog is shown below.
2. Set the **Save Auto-Recovery info** checkbox as necessary. This should default to on.
3. Select the number of minutes you wish the auto-backup facility to be launched.

Restoring your files:

If you open your workspace and the following dialog is displayed it means that the last time the workspace was used problems were encountered.



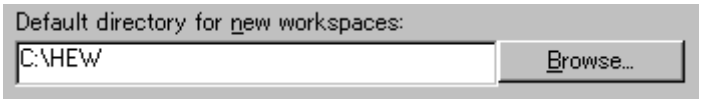
To recover the files check the checkbox alongside the filenames you wish to recover and click OK. Clicking cancel will discard the auto-recovery files and load from the original files.

6.8.8 Default directory for new workspaces

When a new workspace is created, the High-performance Embedded Workshop invokes the **New Project Workspace** dialog. One of the fields on this dialog is the directory in which the new workspace will be created. By default, this is the root directory. However, it is also possible to set this default directory to another location (e.g. '`C:\Workspaces`').

To change the default directory for new workspaces:

1. Select the [Setup–>Options...] menu option. The **Setup Options** dialog will be displayed. Select the **Workspace** tab.
2. Enter the directory in which to create new workspaces into the **Default directory for new workspaces** field, or browse to it graphically by clicking the **Browse** button.
3. Click the **OK** button to confirm the new settings.



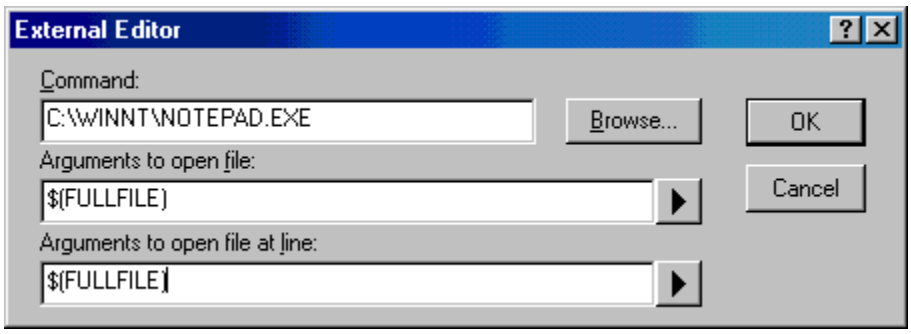
6.9 External editor

The High-performance Embedded Workshop allows you to use an external editor. Once an external editor has been specified, it will be launched when the following actions are performed:

- Double-clicking on a file in the **Projects** tab of the **Workspace** window.
- Double-clicking on an entry in the **Navigation** tab of the **Workspace** window.
- Double-clicking on an error/warning in the **Build** tab of the **Output** window.
- Double-clicking on an entry in the **Find in Files** tab of the **Output** window.
- Selecting the **Open <file>** option from the **Workspace** window's pop-up menu.

To specify an external editor:

1. Select the [Setup–>Options...] menu option. The **Options** dialog will be displayed. Select the **Editor** tab.
2. Check the **Use external editor** checkbox. The **External Editor** dialog will be displayed.



3. Enter the path of the executable (without any arguments) into the **Command** field.
4. Enter the arguments required to open a file into the **Arguments to open file** field. Use the `$(FULLFILE)` placeholder to represent the path of the file to be opened.
5. Enter the arguments required to open a file at a specific line into **Arguments to open file at line** field. Use the `$(FULLFILE)` placeholder to represent the path of the file to be opened and the `$(LINE)` placeholder to represent the line number at which the cursor should be initially positioned. See the Placeholders topic for further information on placeholders.
6. Click the **OK** button to define the editor.

Note:

When using an external editor be aware of the following issues:

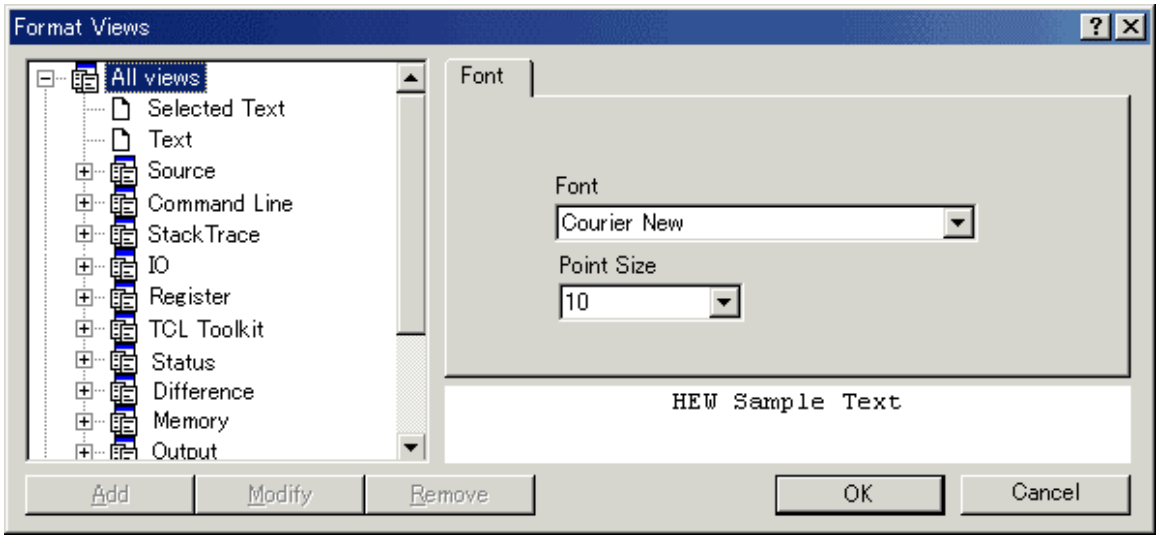
- Each time you invoke the external editor, in whichever way, a separate instance of the editor will be launched.
- You must save your own files before you perform a Build, Build All or Build File operation.

6.10 Customizing the font in your views

The HEW contains many components which you may wish to make look differently. It is possible to change the font and text colouring for the views.

To change the look of your windows:

1. Click the [Setup->Format Views...] menu item.
2. Select the view you are interested in changing the appearance of. To change all views select the "All views" category.
3. Expand the item in the tree to see all items you can change the look of.
4. Select the item. Notice the tab changes on the right of this dialog.
5. Change the font or text colour.
6. Click OK to save the changes and the views will be automatically updated with the new colouring.



6.11 Using the virtual desktop

HEW has implemented the concept of the virtual desktop. This allows window configurations to be defined that can be switched with the click of a button. When a particular button is clicked the windows are hidden or displayed depending on the current settings of that window configuration.

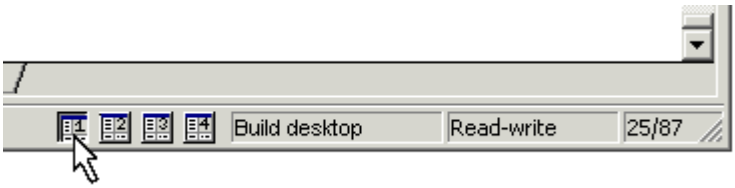
It is possible to have a maximum of 4 desktop configurations in use. When the session is saved the window positions for each configuration are saved to the session file. You can then switch simply between each configuration to gain access to the other windows. The toolbars and windows are dependent on the virtual desktop configuration. Source files are independent of the virtual desktop system and will remain in view.

To rename your configuration to a more meaningful name:

1. Click the [Window->Virtual Desktop] menu item. Select is cascaded menu.
2. Select the Desktop Manager dialog.
3. Select the window configuration you wish to change the name for.
4. Click rename. Enter the new meaningful name in the edit field and click OK.
5. Click OK to keep the changes and revert to the HEW main window.

To switch desktop configurations:

There are a number of ways to switch desktop configuration. The first and easiest method is using the virtual desktop buttons located on the status bar. These are shown below.



In this example the selected desktop is number 1. This has been given the name "Build" by the user. Its description is seen in the edit box to the right on the buttons. Clicking a different desktop selects that button and changes the description control. Once clicked HEW then loads the windows in the new configurations style.

Another method of changing the desktop configuration is as follows:

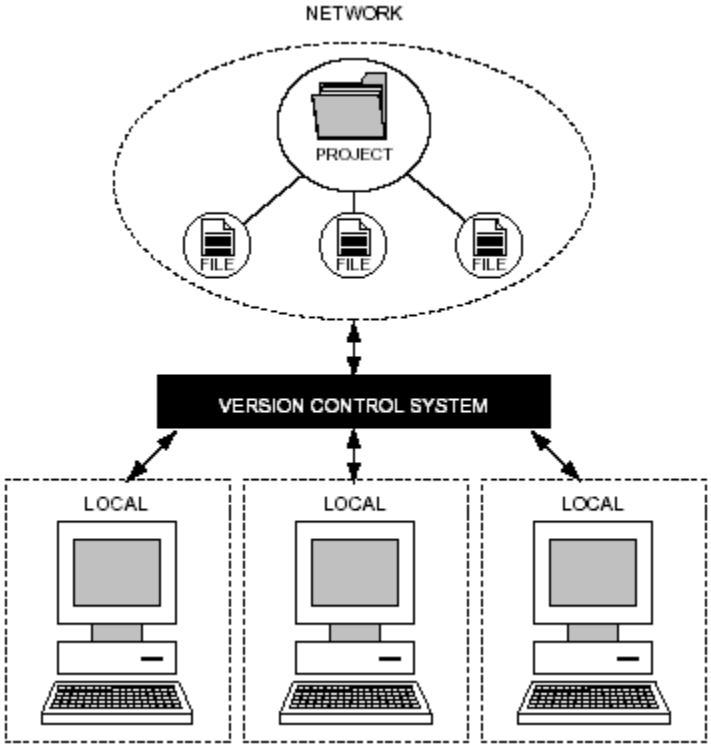
1. Click the [Window->Virtual Desktop] menu item. Select is cascaded menu.
2. Then select the desktop configuration you wish to view on this menu. The selected item is ticked.
3. Select the menu item and the setup is altered automatically.

7. Version Control

The High-performance Embedded Workshop provides facilities for connecting to version control tools. Some of the reasons why version control tools are used with a project are:

- To maintain the integrity of a project.
- To store each stage of a project.
- To enable different users to co-develop a project by controlling revisions to its source files.

Figure below illustrates a typical project where a version control system is in use. This shows three users who all use the same-shared network drive to exchange source code. The version control system provides access and updates to the source files.

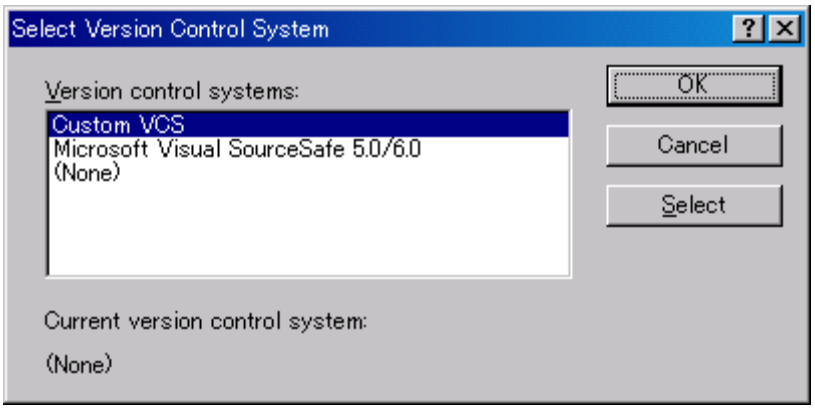


7.1 Selecting a Version Control System

The [Tools→Version Control] sub-menu contains eight menu options but only the [Tools→Version Control→Select...] option is initially available. This is because a version control system is not yet active for the current workspace.

To select a version control system:

1. Select the [Tools→Version Control→Select...] menu option. The [Select Version Control System] dialog will be displayed, which lists all of the supported version control systems.
2. Select the desired version control system from the [Version control systems] list and click the [Select] button. The [Current version control system] field is changed to reflect the new selection.
3. Click the [OK] button to confirm the selection.



Once a version control tool has been selected you will notice that the other menu options in the [Tools→Version Control] sub-menu have now become available.

Note:

Only those version control systems that have been installed with the HEW will appear in the [Select Version Control System] dialog.

Once a version control tool is selected you will notice that the [Version Control->Configure…] option has now become available.

8. Using the Custom Version Control System

The custom version control system is a configurable addition to the High-performance Embedded Workshop, which allows you to connect to a version control system that is already installed on your machine. To clarify further, the High-performance Embedded Workshop does not provide a version control tool itself, only a means for you to integrate the version control system that you use into your workspaces and projects.

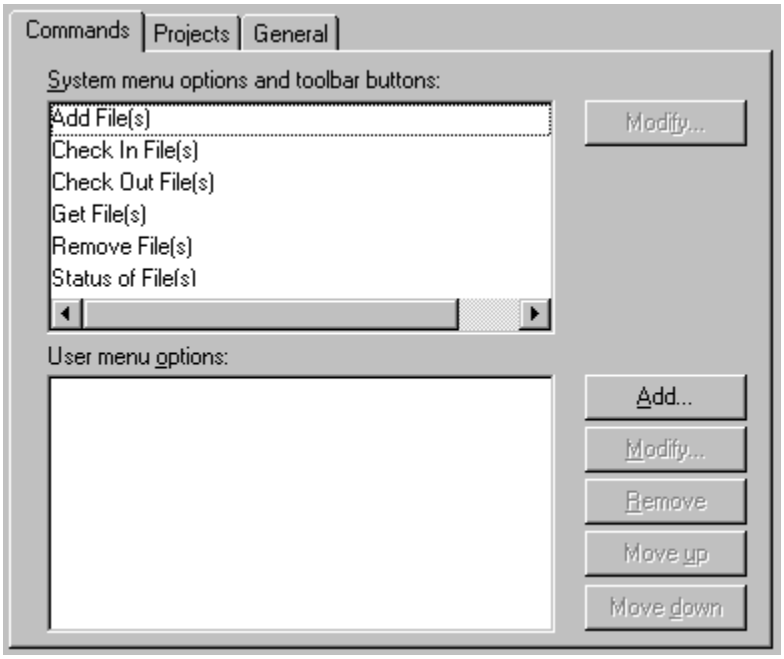
8.1 Defining Version Control Menu Options

The custom version control system allows you to invoke a version control command, either by selecting an option from the [Tools→Version Control] sub-menu, or by clicking a version control toolbar button. When either of these actions are performed, the associated commands are executed and the output is displayed in the [Version Control] tab of the [Output] window.

To execute a version control menu option or toolbar button:

1. Select the items to which you want to apply the version control command, from the [Workspace] window. This may include a workspace, project(s), folder(s) and file(s). When the command is selected, all of the files will be extracted from the selected items and passed, in turn, to the version control command. For example, if you select the workspace icon, all of the files in all of the projects will be passed to the version control command (this will include any system files).
2. Select the required menu option from the [Tools→Version Control] sub-menu or click the desired version control toolbar button.

The custom version control support allows you the highest degree of flexibility in specifying how a version control system is to be used. To configure it, select the [Tools→Version Control→Configure...] menu option. The [Version Control Setup] dialog will be displayed.



The [Commands] tab contains two lists of menu options. The first list, [System menu options and toolbar buttons], represents the menu options that always appear on the [Tools→Version Control] sub-menu. These menu options also have an associated toolbar button on the version control toolbar. The second list, [User menu options], represents those additional user defined options which are added to the bottom of the [Tools→Version Control] sub-menu.

8.1.1 System menu options and toolbar buttons

There are six version control toolbar buttons. They provide you with a shortcut to the most commonly used version control commands. Initially, when you first create a workspace, these buttons are inactive because you have not yet associated any version control commands to them.

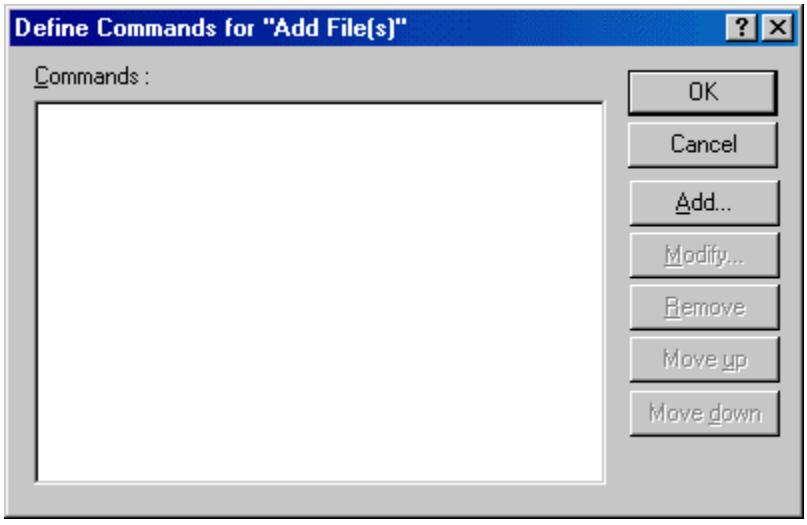
The toolbar buttons are equivalent to the six menu options on the [Tools→Version Control] sub-menu. In other words, selecting the [Tools→Version Control→Get File(s)] menu option will have exactly the same effect as clicking the Get File(s) toolbar button. As the toolbar buttons themselves are fixed, the only operation that you can perform upon them is to define which commands should be executed when they are clicked.

In order to invoke commands from the toolbar or the system defined options of the [Tools->Version Control] sub-menu, you must first define the associated commands that should be executed when they are activated. The names of the options and their intended action are listed.

Option	Description
Add File(s)	Add selected file(s) to version control system.
Remove File(s)	Remove selected file(s) from version control system.
Get File(s)	Get a read only local copy of the selected file(s) from version control system.
Check In File(s)	Put back, i.e. update, the selected file(s) in version control system with the local copy.
Check Out File(s)	Get a writable local copy of the selected file(s) from version control system.
Status of File(s)	View the status of the selected file(s).

To modify the commands associated with a version control toolbar button:

1. Select the [Tools→Version Control→Configure...] menu option. The [Version Control Setup] dialog will be displayed.
2. Select the option to be modified from the top list and then click the [Modify...] button. The [Define Commands] dialog will be displayed.
3. Commands are added via the [Add...] button. See section 8.2, Defining Version Control Commands, for further information.
4. Close the [Define Commands] dialog by clicking the [OK] button.
5. Close the [Version Control Setup] dialog by clicking the [OK] button.

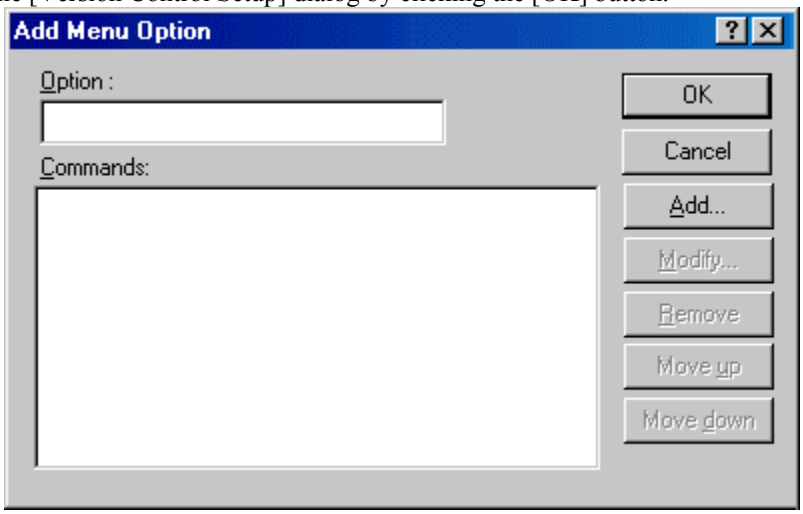


8.1.2 User menu options

You can create as many user-defined menu options as you like, name them how you want and define their order in the menu. User-defined menu options do not appear on the version control toolbar.

To create a new user-defined version control menu option:

1. Select the [Tools→Version Control→Configure...] menu option. The [Version Control Setup] dialog will be displayed.
2. Click the [Add...] button. The [Add Menu Option] dialog will be displayed.
3. Enter the name of the menu option into the [Option] field.
4. To add a command, click the [Add...] button. See section 8.2, Defining Version Control Commands, for further information about adding a command.
5. Close the [Add Menu Option] dialog by clicking the [OK] button.
6. Close the [Version Control Setup] dialog by clicking the [OK] button.



To modify a user-defined version control menu option:

1. Select the [Tools→Version Control→Configure...] menu option. The [Version Control Setup] dialog will be displayed.
2. Select the menu option to be modified from the [User menu options] list and then click the [Modify...] button. The [Modify Menu Option] dialog will be displayed.
3. Modify the commands as necessary and then click [OK] button.
4. Close the [Version Control Setup] dialog by clicking the [OK] button.

To remove a user-defined version control menu option:

1. Select the [Tools→Version Control→Configure...] menu option. The [Version Control Setup] dialog will be displayed.
2. Select the menu option to be removed from the [User menu options] list and click the [Remove] button.
3. Close the [Version Control Setup] dialog by clicking the [OK] button.

To change the ordering of user-defined version control menu options:

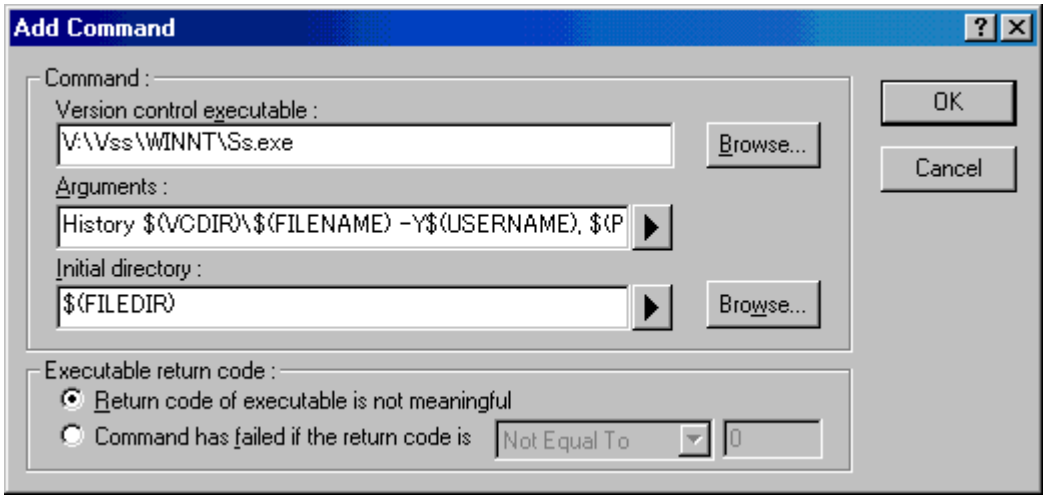
1. Select the [Tools→Version Control→Configure...] menu option. The [Version Control Setup] dialog will be displayed.
2. Select the menu option to be moved and then click the [Move up] and [Move down] buttons as necessary.
3. Close the [Version Control Setup] dialog by clicking the [OK] button.

8.2 Defining Version Control Commands

Version control commands are listed in the [Define Commands] dialog. You can define as many commands as you want to and specify the order in which they execute. Existing commands can be modified or removed.

To define a new version control command:

1. Click the [Add...] button on the [Define Commands] dialog. The [Add Command] dialog will be invoked.



2. Enter the full path of the command into the [Version Control Executable] field, or browse to it graphically by clicking the [Browse...] button.
3. Enter the arguments for the command into the [Arguments] field.
4. Enter into [Initial directory] the directory from which you would like to run the executable or browse to it graphically by clicking [Browse...]. In most cases this should be set to the `$(FILEDIR)` placeholder, which means that the command should be executed from the same directory as the file.
5. Set the [Executable return code] options as appropriate (see below).
6. Click the [OK] button to define the new command.

To modify a version control command:

1. Select the command to be modified from the [Commands] list of the [Define Commands] dialog.
2. Click the [Modify...] button. The [Modify Command] dialog will be displayed.
3. Modify the information as necessary and then click the [OK] button.

To remove a version control command:

1. Select the command to be removed from the [Commands] list of the [Define Commands] dialog.
2. Click the [Remove] button.

To change the ordering of version control menu options:

1. Select the menu option to be moved from the [Commands] list of the [Define Commands] dialog.
2. Click the [Move up] and [Move down] buttons as necessary.

8.2.1 Executable return code

While each version control command executes, its output is redirected to the [Version Control] tab of the [Output] window. When the command's execution is complete, its return code is obtained. When defining a command, you can determine whether this return code can be used to indicate that the command failed and that the remaining commands should not be executed (i.e. abort). The controls contained in the [Executable return code] group allow you to specify this behavior.

If the return code of the command(s) can be used to indicate a failure then you should select the [Command has failed if the return code is] radio button and set the drop-down list and edit box as required. If the [Command has failed if the return code is] radio button is selected then the HEW will check the return code of each command to determine whether a failure occurred. If this is the case, no further commands will be executed and any other processes which would follow the commands (e.g. build) will not be executed.

If the [Return code of tool is not meaningful] option is selected then the HEW will not check the return code of each command. Consequently, all commands will execute regardless.

8.3 Specifying Arguments

It is obvious that arguments must be specified correctly, otherwise the version control tool executed will not function as intended. However, it is also important, when using custom version control support, to specify the arguments in a flexible way, as a single version control command can be applied to more than one file. To facilitate this, the [Arguments] field has a placeholder button (see Reference 4, Placeholders, for an in depth discussion of placeholders), which when clicked on, invokes a pop-up menu of all available

placeholders. An explanation of each version control placeholder and how their values are derived can be found in the table below.

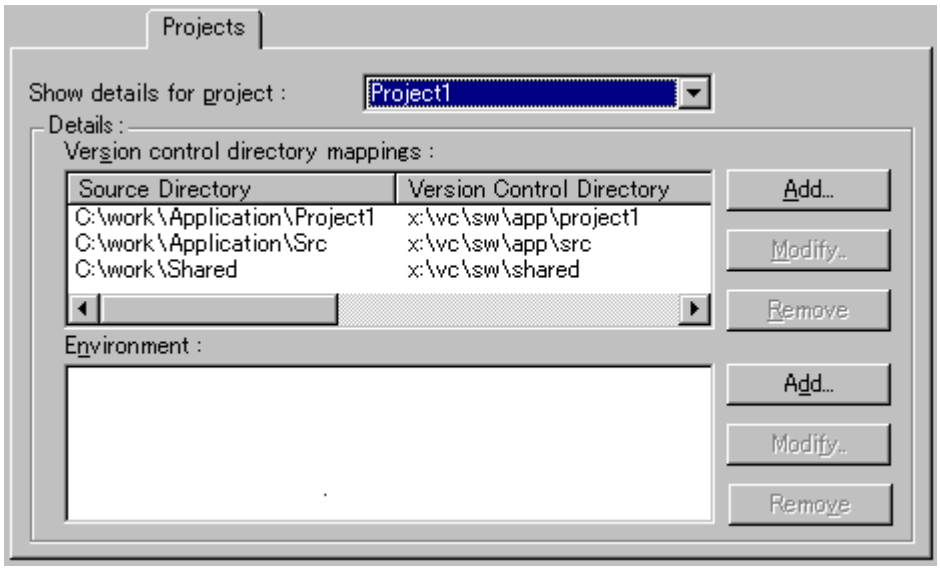
Placeholder	Value And How It Is Determined
User login name	Current user login ('General' tab)
User login password	Current user password ('General' tab)
Version control directory	'Virtual' version control mapping ("Projects" tab)
Comment	Comment specified before command execution
File path + name	Full path and name of the file involved in the operation
Filename	Filename (including extension) of the file involved in the operation
File leaf	Filename (excluding extension) of the file involved in the operation
File extension	Extension of the file involved in the operation
File directory	Directory of the file involved in the operation
Configuration directory	Current configuration directory
Project directory	Current project directory
Workspace directory	Current workspace directory
Temp Directory	Temporary directory
Command directory	Version control executable directory
Windows directory	Directory where Windows® is installed
Windows system directory	Directory where Windows® system files exist
Workspace name	Current workspace name
Project name	Current project name
Configuration name	Current configuration name

8.3.1 Specifying File Locations

When referring to a file's location, be sure to use a placeholder, otherwise the command will only relate to a hardwired file. For example, let's imagine that a version control executable has been selected which uses a `-GET` command to obtain a read-only copy of a file. The [Arguments] field could be specified as:

```
-GET 'c:\vc\files\project\main.c'
```

However, when executed, this command can only ever `-GET` the file `MAIN.C`. To resolve this problem, HEW uses a system of placeholders and directory mappings. Directory mappings tell the HEW which 'working' directories (i.e. where source files are being worked on) map to which 'controlled' directories (i.e. where the source files are stored in the version control system). Mappings between these two directory systems can be specified via the [Projects] tab of the [Version Control Setup] dialog.

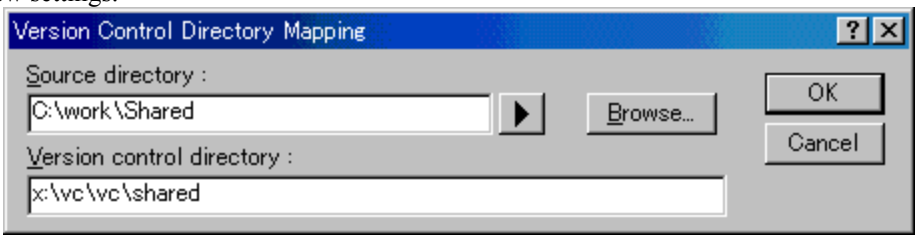


To define a new directory mapping:

1. Select [Tools→Version Control→Configure...]. The [Version Control Setup] dialog will be displayed. Select the [Projects] tab.
2. Click the [Add...] button, which is next to the [Version Control Directory Mappings] list. The [Version Control Directory Mapping] dialog will be displayed.
3. Enter the source directory (i.e. 'working' directory) into the [Source Directory] field, or browse to it graphically by clicking the [Browse...] button.
4. Enter the version control directory (i.e. 'controlled' directory) into the [Version Control Directory] field.
5. Click the [OK] button to define the new mapping.

To modify a directory mapping:

1. Select [Tools→Version Control→Configure...]. The [Version Control Setup] dialog will be displayed. Select the [Projects] tab.
2. Select the mapping to be modified from the [Version Control Directory Mappings] list and then click the [Modify...] button. The [Version Control Directory Mapping] dialog will be displayed.
3. Make the necessary changes to the two directories and then click the [OK] button to confirm the new settings.

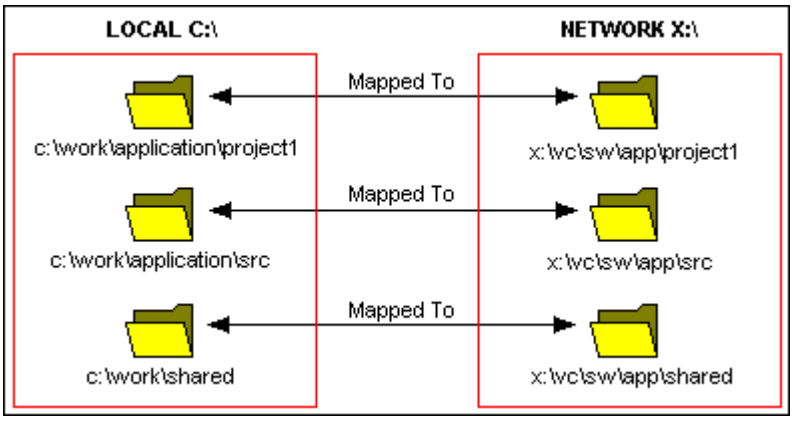


To remove a directory mapping:

1. Select [Tools→Version Control→Configure...]. The [Version Control Setup] dialog will be displayed. Select the [Projects] tab.
2. Select the mapping to be removed from the [Version Control Directory Mappings] list and then click the [Remove] button.

8.3.2 Specifying File Locations Example

Consider the scenario shown in the figure below. It shows three directories, which are mapped from a shared version control drive (`X:\`) to a local drive where the development is being done (`C:\`).



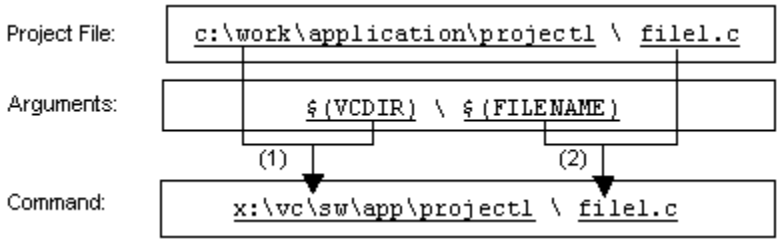
Now let's imagine that a version control executable has been selected which uses a `-GET` command to obtain a read-only copy of a file. In order to get all of the files in a project we need to use the following command:

```
-GET '$(VCDIR)\$(FILENAME)'
```

When the HEW executes the command for a given project file, it will replace `$(VCDIR)` for the equivalent version control directory in the file mapping.

For example, suppose `FILE1.C` is located at `c:\work\application\project1\FILE1.C`. If the `-GET` command is applied to `FILE1.C` then:

1. `'x:\vc\sw\app\project1'` is substituted for `'$(VCDIR)'`, as this is the version control directory mapping for `'c:\work\application\project1'`.
2. `'FILE1.C'` is substituted for `'$(FILENAME)'`.

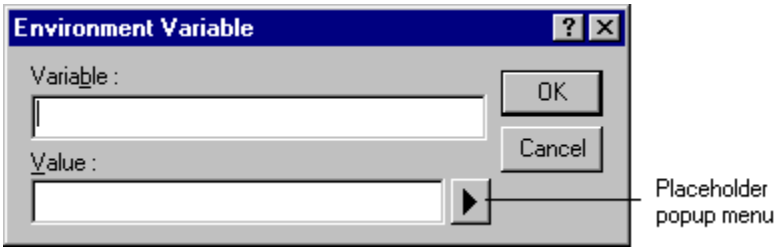


8.3.3 Specifying Environment

Select the [Projects] tab of the [Version Control Setup] dialog to view the current settings.

To add a new environment variable:

1. Click the [Add...] button beside the [Environment] list (the [Environment Variable] dialog will be invoked).



2. Enter the variable name into the [Variable] field.
3. Enter the variable's value into the [Value] field.
4. Click the [OK] button to add the new variable to the Environment list.

To modify an environment variable:

1. Select the variable that you want to modify from the [Environment] list.
2. Click the [Modify...] button beside the list.
3. Make the required changes to the [Variable] and [Value] fields.
4. Click the [OK] button to add the modified variable back to the list.

To remove an environment variable:

1. Select the variable that you want to remove from the [Environment] list.
2. Click the [Remove] button beside the list.

8.3.4 Specifying Comments

If a version control command contains the placeholder '`$(COMMENT)`', the HEW will request that you enter the comment when the command is executed (via the [Please Enter Comment] dialog).

You may specify a comment for each file or, if you would like to specify the same comment for all files, check the [Apply comment to all files] checkbox before clicking the [OK] button.

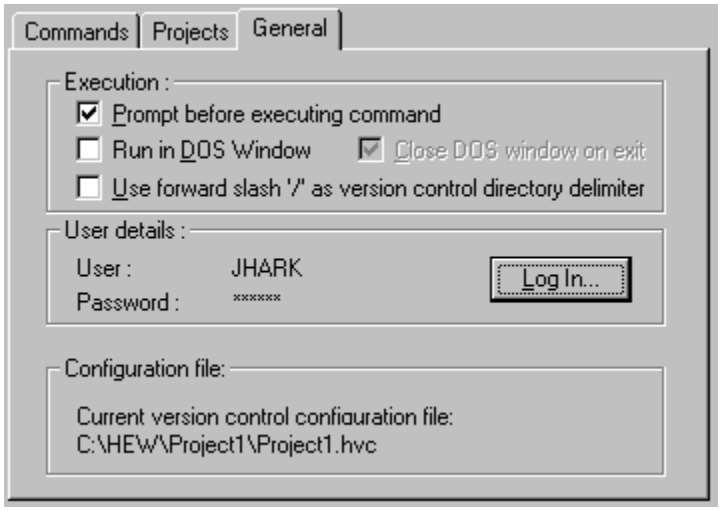
8.3.5 Specifying a User Name and Password

Most version control tools will require you to pass a username and password on the command line in order to keep files secure, and to keep a record of which files were changed by which users. The custom version control support provides two placeholders: **User Login Name**, `$(USERNAME)`, and **User Login Password**, `$(PASSWORD)`. When the command is executed, these placeholders will be replaced with the current settings in the [General] tab of the [Version Control Setup] dialog.

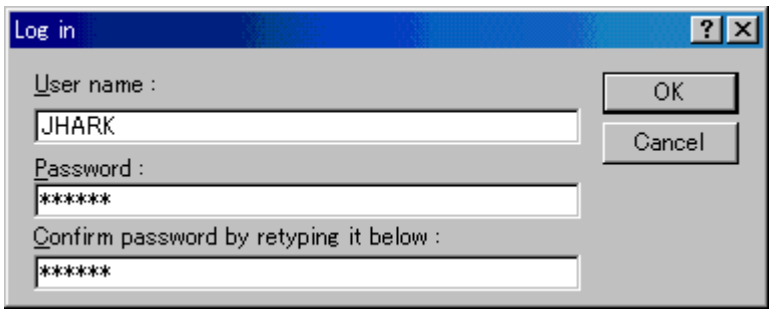
In order to give the `$(USERNAME)` and `$(PASSWORD)` fields a value you will first need to login. If you have not logged in before a command is executed, which uses either of these placeholders, then you will be prompted to do so before the command can be executed.

To login (i.e. specify a username and password):

1. Select the [Tools→Version Control→Configure...] menu option. The [Version Control Setup] dialog will be displayed. Select the [General] tab.



2. Click the [Log In...] button. The [Log in] dialog will be displayed.
3. Enter your username into the [User name] field.
4. Enter your password into the [Password] field.
5. Re-type your password again into the [Confirm password by retyping it below] field.
6. Click the [OK] button to set the new username and password. If there is any inconsistency between the two versions of the password that you entered, you will be requested to type your password again.



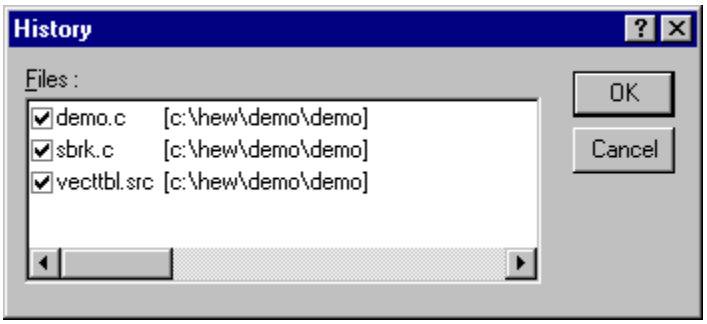
8.4 Controlling Execution

The [General] tab of the [Version Control Setup] dialog allows you to control the way in which the version control tool is executed. It also shows the full path to the current version control configuration file.

The execution of a version control tool can be modified via the following three checkboxes:

Prompt before executing command

If this checkbox is set then, before any version control commands are executed, a dialog is displayed, which lists all of the files involved in the operation. Files may be deselected by clearing the associated checkbox. Clicking the [OK] button will apply the command to each of the selected files. Clicking the [Cancel] button will abort the operation.



Run in DOS window

By default, the output of the version control commands is redirected to the [Version Control] tab of the [Output] window. If you would rather run each command in a separate DOS window then set this checkbox.

Use forward slash '/' as version control directory delimiter

By default, when the HEW substitutes the `$(VCDIR)` placeholder, it uses the backward slash character '\' to divide directories. However, if the version control system you are using uses a forward slash character (e.g. Visual SourceSafe) to divide directories then set the [Use forward slash '/' as version control directory delimiter] checkbox.

8.5 Importing and exporting a Set-up

Each workspace can have a different version control setup. The HEW allows you to store the version control settings independently so that you can import them into other workspaces. This greatly reduces the amount of time it takes to configure the same version control settings across several workspaces.

To export a version control setup:

1. Select the [Tools→Version Control→Configure...] menu option. The [Version Control Setup] dialog will be displayed.
2. Click the [Export...] button. A [File Save] dialog will be displayed.
3. Browse to the directory in which you would like to save the configuration.
4. Enter the name of the file and then click the [OK] button.

To import a version control setup:

1. Select the [Tools→Version Control→Configure...] menu option. The [Version Control Setup] dialog will be displayed.
2. Click the [Import...] button. A [File Open] dialog will be displayed.
3. Browse to the `*.HVC` file that you would like to import.
4. Select the file and then click the [OK] button.

9. Using Visual SourceSafe

The High-performance Embedded Workshop provides specific support for the Visual SourceSafe version control system. The Visual SourceSafe version control system associates a project in your workspace with a project inside a Visual SourceSafe database. It allows you to quickly invoke the standard commands either by selecting an option from the [Tools→Version Control] sub-menu or by clicking a version control toolbar button.

9.1 Attaching Visual SourceSafe to a workspace

The following sections describe how you can associate Visual SourceSafe with your current workspace.

9.1.1 Selecting Visual SourceSafe

First, you need to select Visual SourceSafe as the version control system.

To use Visual SourceSafe 5.0 or 6.0:

1. Select the [Tools→Version Control→Select...] menu option. The [Select Version Control System] dialog will be displayed, which lists all of the supported version control systems.
2. Select the [Visual SourceSafe 5.0/6.0] entry from the [Version Control Systems] list and click the [Select] button.
3. Click the OK button to confirm the selection. The [SourceSafe Login] dialog is displayed.
4. Enter your Visual SourceSafe username into the [Username] field and password into the [Password] field.
5. Enter into [Database path] the full path to the Visual SourceSafe database (i.e. `SRCSAFE.INI`) into which you would like to add this project. Alternatively, you can locate the folder graphically by clicking the [Browse] button.
6. Click the OK button. The [Create SourceSafe Project] dialog is invoked.
7. The [Project name] field displays the name of the project (i.e. folder) to be created in the database. If necessary you can change this name to another.
8. The tree underneath the [Project name] field shows the structure of the database specified in Step 5. Select the folder into which you would like to create the folder specified in [Project name].
9. Click the [OK] button.
10. HEW will require you to repeat steps 7-9 for as many projects as are present in the current workspace.

The High-performance Embedded Workshop has now created the necessary projects within Visual SourceSafe, and set up the version control toolbar and menu for immediate access. However, although the Visual SourceSafe projects themselves have been created, no files have been added to them.

9.1.2 Adding files to Visual SourceSafe

There may exist a mapping between the project directory on your hard disk (i.e. the working directory) and the project directory in Visual SourceSafe (i.e. the controlled directory) (see the Directory mappings topic for further information). However, the project directory (and any subdirectories) on your hard disk may contain many source files, whereas the directory it is mapped to in Visual SourceSafe will be initially empty. You must tell the HEW which files should be controlled, i.e. which files should be added to the Visual SourceSafe project.

To add a file(s) to Visual SourceSafe:

1. Ensure that Visual SourceSafe has been selected as the version control system (see the Connecting a workspace to Visual SourceSafe topic for further information).
2. Select the file(s) that you would like to add to Visual SourceSafe, in the [Workspace] window. You may also select a file folder, project folder, workspace folder or combination thereof. When selecting the project or workspace folder, the system files will be added to the selected file list. For example, selecting the project folder will also add the project file to the file list. If the project file is then checked out and the version is newer than when it was last loaded, you will be asked whether you want to reload the project.
3. Click the Add Files toolbar button  , or select the [Tools→Version Control→Add Files] menu option.

When you add files to Visual SourceSafe, the local versions in your working directory will become read-only. To check that the **Add Files** operation was carried out as you expected, or to quickly review the status of all of the files in a project:

1. Select the project folder whose files you want to check, in the [Workspace] window.
2. Click the Status of Files toolbar button  , or select the [Tools→Version Control→Status of Files] menu option.
3. The status of each file will be displayed in the [Version Control] tab of the [Output] window. The information shown includes whether the file is added to the project, if the file is checked out and, if it is checked out, who did so.

9.2 Visual SourceSafe commands

The following 8 operations are available:

- Adding files to version control
- Remove a file from version control
- Get a read only copy of a file or files
- Check out a read/write copy of a file or files (i.e. for editing)
- Check in a previously checked out file or files (i.e. update Visual SourceSafe with the edits made)
- Undo a previously check out operation on a file or files (i.e. cancel any edits made)*
- View the status of a file
- View the history of a file *

* These commands can only be accessed via the [Tools→Version Control] sub-menu, whereas all of the other commands can be accessed from both the toolbar and the menu.

9.2.1 Removing a File from Version Control

To remove a file(s) from Visual SourceSafe:

1. Select the file(s) that you would like to remove from Visual SourceSafe, in the [Workspace] window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Click the Remove Files toolbar button  , or select the [Tools→Version Control→Remove Files] menu option.

9.2.2 Getting a Read Only Copy of a File from Version Control

Visual SourceSafe protects your source files and ensures that only one user can have a writable copy of a controlled file at any one time. However, it is possible for any user to obtain a read-only copy of any file.

To get a read-only copy of a file(s) from Visual SourceSafe:

1. Select the file(s) that you would like to get from Visual SourceSafe, in the [Workspace] window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Click the Get Files toolbar button  , or select the [Tools→Version Control→Get Files] menu option.

9.2.3 Checking Out a Writable Copy of a File from Version Control

Visual SourceSafe protects your source files and ensures that only one user can have a writable copy of a controlled file at any one time. The check out operation takes a writable copy of the file from Visual SourceSafe and places it on your local drive. This can only be done if another user has not already checked out the file(s) in question.

To check out a writable copy of a file(s) from Visual SourceSafe:

1. Select the file(s) that you would like to check out from Visual SourceSafe, in the [Workspace] window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Click the Check Out Files toolbar button  , or select the [Tools→Version Control→Check Out Files] menu option.

9.2.4 Checking In a Writable Copy of a File into Version Control

Visual SourceSafe protects your source files and ensures that only one user can have a writable copy of a controlled file at any one time. The check out operation takes a writable copy of the file from Visual SourceSafe and places it on your local drive. Once a file is checked out it is edited and then checked back in so that the edits can be made available to other users.

To check in edits made to a file(s) in Visual SourceSafe:

1. Select the file(s) that you would like to check back into Visual SourceSafe. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Click the Check In Files toolbar button  , or select the [Tools→Version Control→Check In Files] menu option.

9.2.5 Undoing a Check Out Operation

Visual SourceSafe protects your source files and ensures that only one user can have a writable copy of a controlled file at any one time. The check out operation takes a writable copy of the file from Visual SourceSafe and places it on your local drive. Once a file is checked out it is edited and then checked back in so that the edits can be made available to other users. However, if the check out operation was carried out by mistake, or perhaps is no longer required, then the operation can be undone.

To undo a check out of a file(s) from Visual SourceSafe:

1. Select the file(s) upon which you would like to undo a previous check out operation. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Select the [Tools→Version Control→Undo Check Out] menu option.

9.2.6 Viewing the Status of a File

Although files appear in your HEW project (in the **Projects** tab of the **Workspace** window), Visual SourceSafe is not necessarily controlling them. Some of the files that are being controlled by Visual SourceSafe will be checked in, and others will be checked out (i.e. being edited by a user). The **Status of Files** command displays the current status of a file or file(s).

To view the status of a file(s) in Visual SourceSafe:

1. Select the file(s) whose status you would like to view, in the [Workspace] window. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Click the Status of Files toolbar button  , or select the [Tools→Version Control→Status of Files] menu option.
3. If a file has a red tick next to it a user different to you has checked it out.

9.2.7 Viewing the History of a File

Visual SourceSafe controls the edits to the files in its projects and allows you to view the complete history of these edits right back to the time that the file was first added to the project.

To view the revision history of a file(s) in Visual SourceSafe:

1. Select the file(s) whose history you would like to view. You may also select a file folder, project folder, workspace folder or combination thereof.
2. Select the [Tools→Version Control→Show History] menu option.

9.3 Visual SourceSafe Integration Options

You can control the way in which the history and status commands are displayed by selecting the [Tools→Version Control→Configure...] menu option.

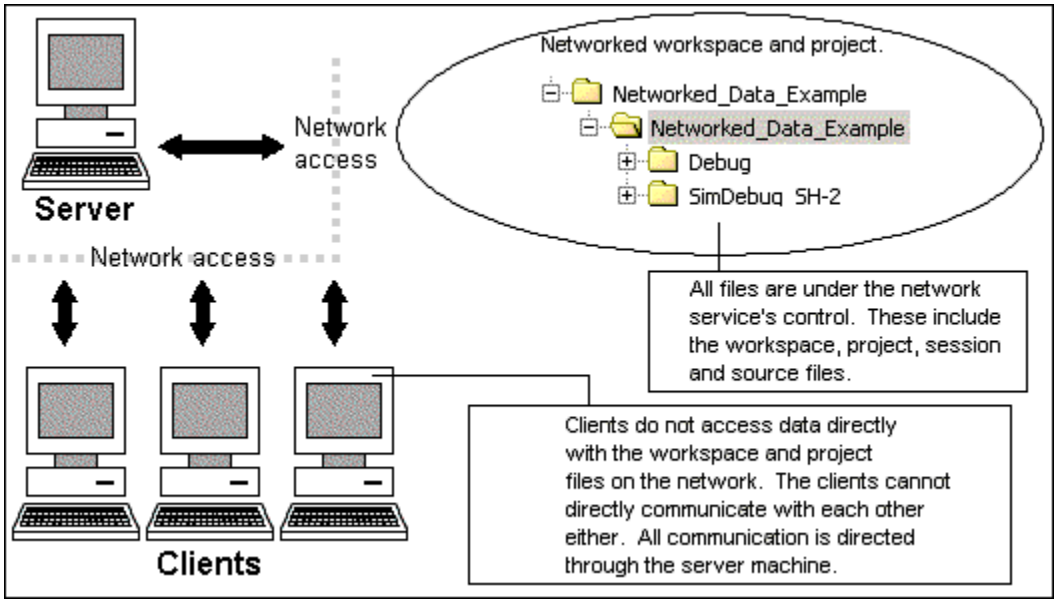
To display the results of a history command in a dialog box then check the [Display dialog box for history] checkbox or clear it if you would rather display the output in the [Version Control] tab of the [Output] window.

To display the results of a status command in a dialog box, check the [Display dialog box for file status] checkbox or clear it if you would rather display the output in the [Version Control] tab of the [Output] window.

10. Network Facilities

The High-performance Embedded Workshop is capable of sharing workspaces and projects across a network. This allows users to concurrently work on shared projects and see each other's changes as they happen. This system can be used in conjunction with version control. The major difference with using this system is that each user can modify and update the workspace and project without making all of the other users reload their project and potentially lose all their changes.

This system is implemented by making one of the machines attached to the network the server machine. All other client machines then use the service this machine is providing. So if one of the client machines adds a new file, the server machine is notified. The server then notifies all other clients the action has taken place. This procedure is shown below.



The network system allows users to be given access rights to files. This allows the project administrator to make sure the only people who can modify the project and source files are allowed to do so. This might allow the administrator to limit each user to only have write capabilities for their own area of the project, other areas would be read only. This could limit any potential conflict or damage one user could do to other areas of the project. These limitations can be set to a number of different levels. This is outlined later in this section.

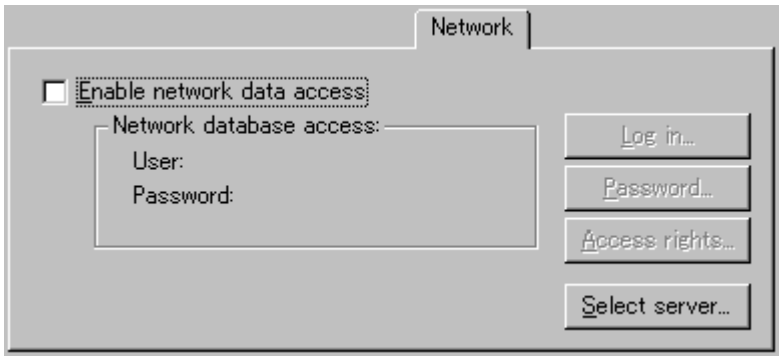
Note:

- Certain operations are locked when other clients are carrying them out. This means that if one machine is currently changing the toolchain options all of the other machines can only see read only versions of this data.
- The performance of HEW does suffer when using the network facilities. If working in a small team it might be more suitable to use the single user mode and version control.

10.1 Enabling network access

To use network access:

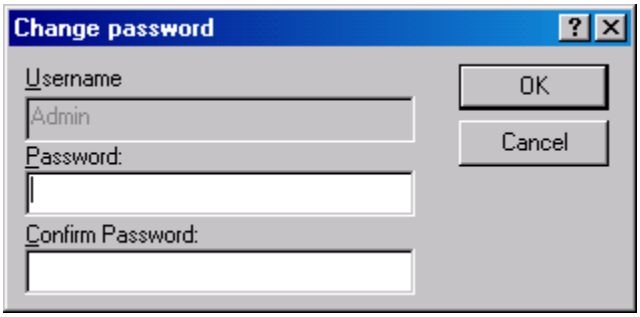
1. Select [Setup->Options...]. The [Options] dialog is displayed.
2. Select the [Network] tab.
3. Click the [Enable network data access] checkbox. This should add an administrator to the system without a password. The administrator is the only user that can add additional users to the system and change user access rights. The administrator has the highest level of access.
4. Before leaving the network dialog the administrator must set their password. It is not possible to leave this dialog until this is completed. This is described below.



10.2 Setting the administrator user's password

To set the administrator users password:

1. Continue from the previous sections steps.
2. Click the [Password...] button. This should have been enabled when the network data access was enabled.
3. The [Change password] dialog is displayed.



4. The user name is read only in the top field. In this case it should be Admin.
5. Type the new password into both of the fields and click [OK].
6. This should set the user and password on the [Network] tab of [Options] dialog.
7. It is now possible to leave the [Options] dialog.
8. When the dialog is closed you are asked if you want to save the workspace and then re-open it. This is because the workspace must be re-opened in the shared access mode. If the changes are not saved then they will be lost.
9. When the workspace is re-opened a dialog is displayed which asks you to log back into the system. Once you have logged in a dialog is displayed which shows your current access rights. For example if you are the admin user the level will be administrator. When this dialog is closed the HEW server window is opened and the network facilities are enabled.

10.3 Adding new users to the system

The initial setting of the network database adds an administrator user and a guest user to the system. The following levels of access are possible in the HEW system:

Administrator	Full access to every aspect of HEW. The user can add and remove users from the projects and change access rights. The administration user can change the workspace and project files and also the source files.
Full read/write access	The workspace and project files can be modified, as can the source files. But it is not possible to change user access rights from this access level.
Read/write file access	Only the source files can be modified. All project settings can only be viewed not modified.
Read only	All source files and project files can only be viewed as read only. Nothing can be modified.

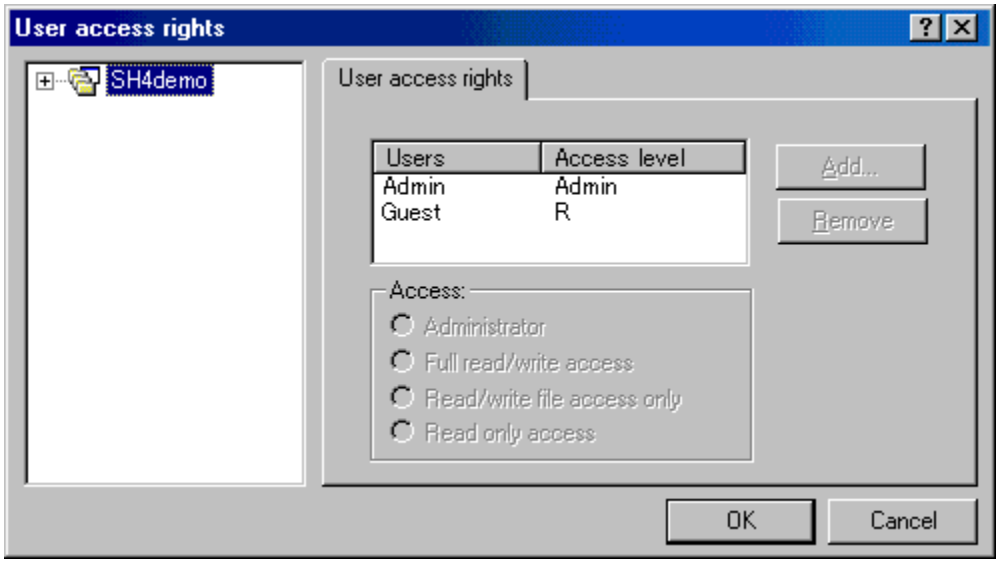
When any user opens a network-enabled project they must type in their user name and password. Until this is done no access can be granted. Once entered the user is given one of the levels of access as seen above.

To add a new user to the system:

1. Log in with a user who has administrator access rights. The process for doing this is described above.
2. Select [Setup->Options...]. The [Options] dialog is displayed.
3. Select the [Network] tab.
4. Click the [Access rights...] button. The [User access rights] dialog is displayed.
5. Click the [Add...] button. The [Add new user] dialog is displayed. This allows you as the administration user to add a new log in name and password. Normally the password should be set to some default text or left blank. Then click [OK].
6. Once [OK] is clicked the user is added with read only rights. To change the access level select the user you wish to modify and then click the required radio button. Then click [OK] to save the access rights changes.

To remove an existing user to the system:

1. Log in with a user who has administrator access rights. The process for doing this is described above.
2. Select [Setup->Options...]. The [Options] dialog is displayed.
3. Select the [Network] tab.
4. Click the [Access rights...] button. The [User access rights] dialog is displayed.
5. Select the user you wish to remove in the users list.
6. Press the [Remove] button.
7. Then click [OK] to save the access rights changes.



10.4 Changing your password

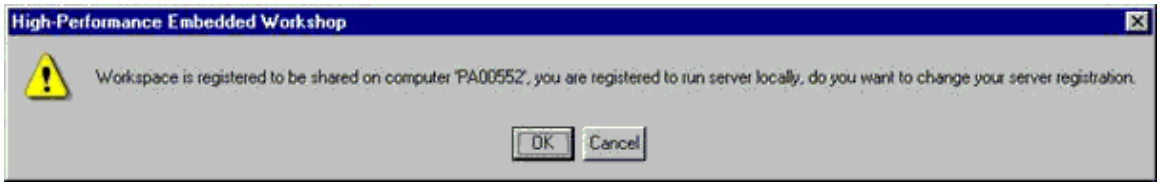
To change your password:

1. Log into the HEW network database you are changing your password for. Select [Setup->Options...]. The [Options] dialog is displayed.
2. Select the [Network] tab.
3. Click the [Password...] button.
4. Enter your new password and confirm it in the second edit box.
5. Click [OK].
6. Then click [OK] to save the password change.

10.5 Using the network HEW service

When you connect to a networked project for the first time the HEW automatically connects you to the correct network HEW service. This is defined using machine name. If the service cannot be found using the machine name in the workspace then the dialog below is shown. Simply type or browse to the machine where the service is located and click [OK]. If you want to be the server machine then leave the radio button on its default selection, use local machine.

If you have previously been the server of a workspace then the following message will be displayed when you attempt connection to another machine. Clicking [OK] then connects your machine to the new location.



Note:

If the network is running multiple HEW workspaces with the network service enabled then a user can only access one of them at one time. The only instance when this is not the case is if the same machine is serving all of the network workspaces.

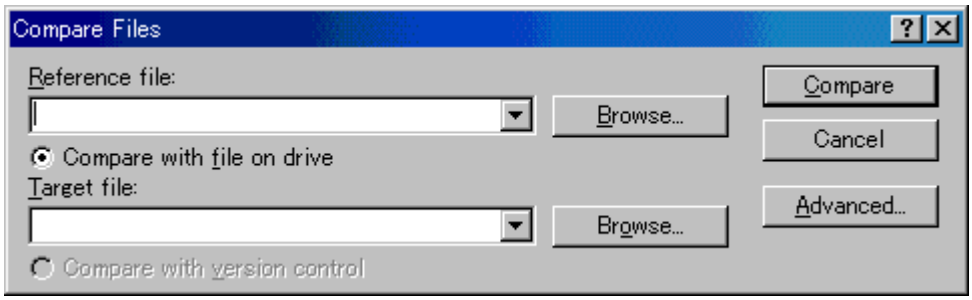
11. Comparing Files

The High-performance Embedded Workshop now has an integrated difference view. This view allows detailed difference comparisons to be made with local files on the drive and also with files in the version control system. In the HEW version 3.0 onwards the Visual SourceSafe component has this facility.

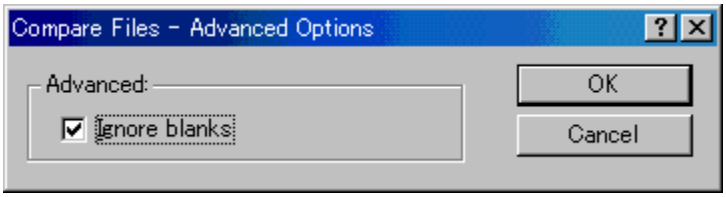
The [Difference] window can be invoked two ways. The first is from the [Show Differences...] menu item on the tools menu. The second is via the workspace window pop-up with the same name. When using the pop-up menu the current file selection is automatically added to the edit box. Clicking the show differences menu item displays the difference view. The view has a toolbar for accessing functionality too.

To perform a difference comparison with two files on your local drive:

1. Select [Tools->Show Differences…]. The [Compare Files] dialog is displayed.



2. Ensure the [Compare with file on drive] radio button is enabled.
3. Enter the first and second file to compare. You can either select a previous difference comparison or browse to a new file.
4. Clicking the [Advanced...] button displays the [Compare Files - Advanced Options] dialog. This allows you to perform the difference comparison without taking white space into account. Click [OK] when you are finished with this options dialog.



5. Click [Compare].

To perform a difference comparison with a local file and a file in SourceSafe:

1. Ensure the SourceSafe component is enabled. Also note that the file must be have been added into the version control system.
2. Select [Tools->Show Differences…]. The [Compare Files] dialog is displayed.
3. Ensure the [Compare with version control] radio button is enabled.
4. Enter the first file to compare. You can either select a previous difference comparison or browse to a new file.
5. Clicking the [Advanced...] button displays the [Compare Files - Advanced Options] dialog. This allows you to perform the difference comparison without taking white space into account. Click [OK] when you are finished with this options dialog.
6. Click [Compare].

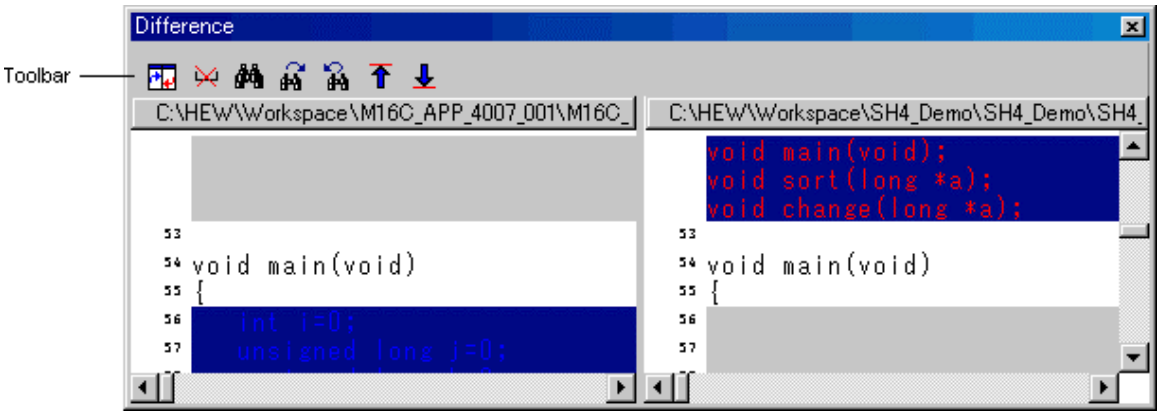
The [Difference] window is displayed.

11.1 Opening the Difference window

Continue from the previous sections steps. The [Difference] window is displayed.

The two files being compared are loaded into each side of the split view. Their names are at the top of each window.

Configuration of Difference window



Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

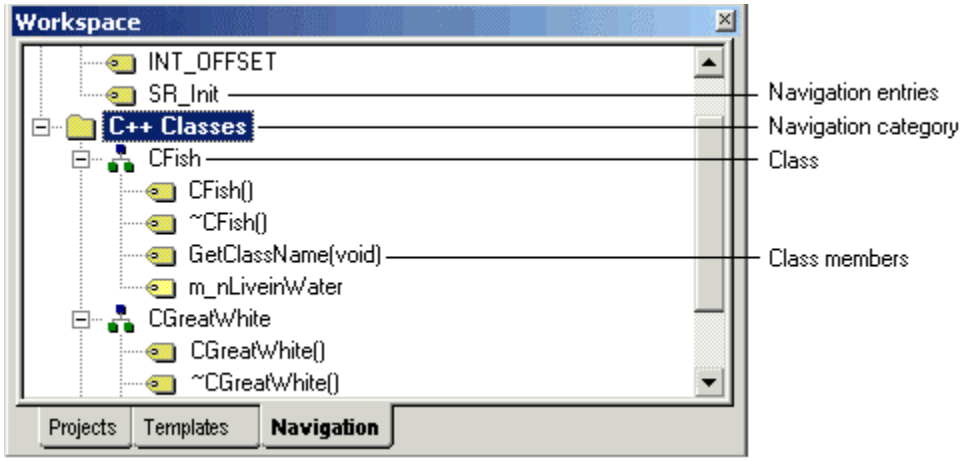
A basic operation is allocated to the toolbar.

The functions of [Toolbar display] and [Customize toolbar...] are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar bottom	Function
Compare...		This opens a new compare window so that some new files can be compared and the differences displayed.
Export results to file	-	This opens a dialog which allows you to choose a file to export the current difference results to a textual format.
Ignore white space		The ignore white space option which is on the advanced options dialog can be toggled via this menu item.
Find		Displays a standard find dialog. This uses the same find dialog as the HEW editor.
Find previous		Finds the next previous string that meets the find requirements.
Find next		Finds the next string that meets the find requirements.
Previous difference		Automatically jumps the view to the next previous difference.
Next difference		Automatically jumps the view to the next difference.
Refresh comparison	-	Refreshes the view to manually run the difference comparison again. This can be used if either file has been modified since the last comparison.
Toolbar display	-	Showing/hiding toolbar buttons
Customize toolbar...	-	Customizing toolbar buttons

12. Navigation Facilities

The High-performance Embedded Workshop has a number of new integrated navigation facilities.



The [Navigation] tab of the workspace window contains categories for all supported navigation types. In HEW the following navigation components are supported as standard:

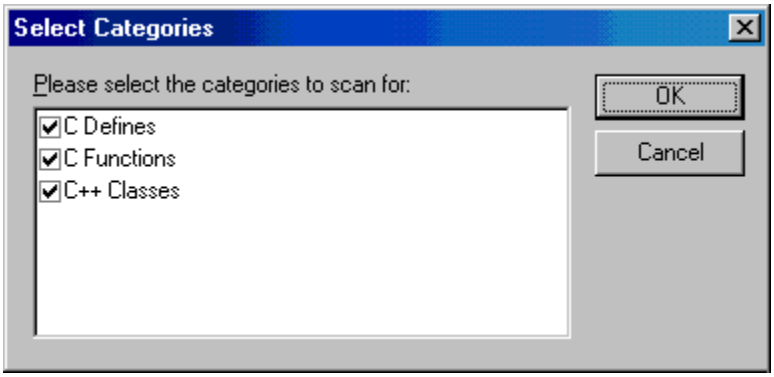
Navigation Type (Category)	Function
C Defines	All #defines for C and C++ source files are displayed.
C Functions	All ANSI C standard functions are for C source files displayed.
C++ Class	All classes, functions and members are displayed for C++ source files.

Each category is displayed in the top level of the view. Underneath each category, the items belonging to the active project are displayed.

It is possible to disable scanning for certain navigation categories if you do not require the information.

To switch off a navigation category:

1. If you right-click anywhere inside the [Navigation] tab, a pop-up menu will be invoked.
2. Select the [Select Categories…] menu item.
3. The [Select Categories] dialog is displayed.



4. Uncheck any categories you are not interested in seeing definitions for.
5. Click [OK].

To update the navigation view:

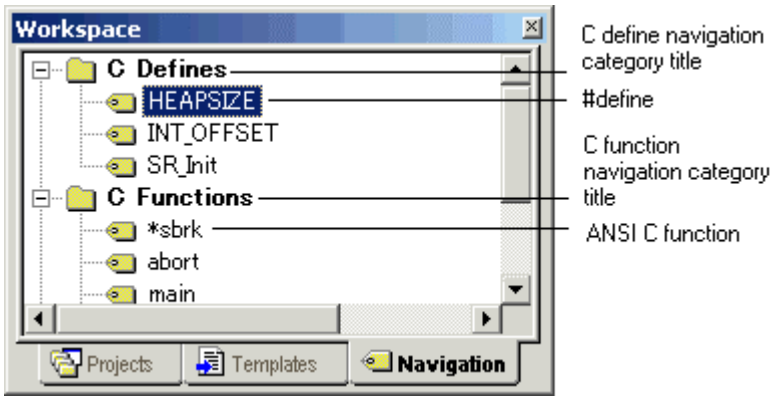
1. If you right-click anywhere inside the [Navigation] tab, a pop-up menu will be invoked.
2. Select [Refresh].

Note:

The navigation items are displayed gradually as the files are scanned. This means it may take some time if there are many files to fully complete the Navigation view update. Files are rescanned when they are saved. This means that navigation information will not be available for new classes and functions until the file or files are saved.

12.1 C Function and #define navigation component

These components simply add the function and #define definitions to the navigation view.



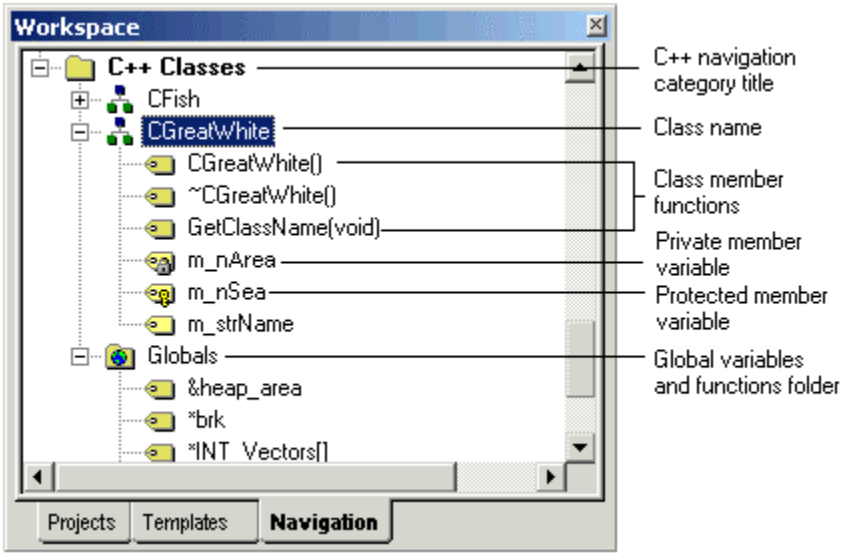
To jump to a definition:

Select either of the following ways.

- Double-click a function or a #define definition on the [Navigation] tab.
- Right-click on a function or a #define definition on the [Navigation] tab. Select [Go to Definition] from the pop-up menu.

12.2 C++ Navigation component

The C++ navigation component is the most complicated of the three supported navigation components. It supports the following structures in the view for C++ source files. The basic structure of the information is shown below.



The C++ navigation view uses a number of icons to describe the type of function or variable the icon belongs too. These are listed in the table below:

Icon	Description
	Public member function
	Protected member function
	Private member function
	Public member variable
	Protected member variable
	Private member variable

Double clicking on a navigation item by default jumps you to the associated navigation items declaration. This default behavior can be modified by selecting [Jump to definition on double click] from the pop-up menu (this option is unchecked at default). When this option is checked, double-clicking a navigation item jumps you to the associated navigation items definition.

To jump to the definition:

Select either of the following ways.

- Right-click on navigation items on the [Navigation] tab to display a pop-up menu and check that [Jump to definition on double click] is checked. Double-click a navigation item on the [Navigation] tab.
- Right-click on navigation items on the [Navigation] tab to display a pop-up menu. Select [Jump to definition].

To jump to the declaration:

Select either of the following ways.

- Right-click on navigation items on the [Navigation] tab to display a pop-up menu and check that [Jump to definition on double click] is unchecked. Double-click a navigation item on the [Navigation] tab.
- Right-click on navigation items on the [Navigation] tab to display a pop-up menu. Select [Jump to declaration].

To list the member variables and functions in the alphabetical order:

1. Right-click on navigation items on the [Navigation] tab to display a pop-up menu.
2. Uncheck [Group by access]. This option is unchecked at default.

To group the display of public, private, and protected member variables and functions together:

1. Right-click on navigation items on the [Navigation] tab to display a pop-up menu.
2. Check [Group by access].

Another useful facility is the capability of viewing the base or derived classes for a certain selection.

To view the Base or derived classes:

1. Right-click on class on the [Navigation] tab to display a pop-up menu.
2. To see the derived classes for the selection click the [Show Derived Classes] menu item. To see the base classes for the selection click the [Show Base Classes] menu item.
3. Depending on the selection a dialog is displayed which shows the class structure selected in an expanded tree format.
4. Click [Close] to close this dialog once you have the information you require.

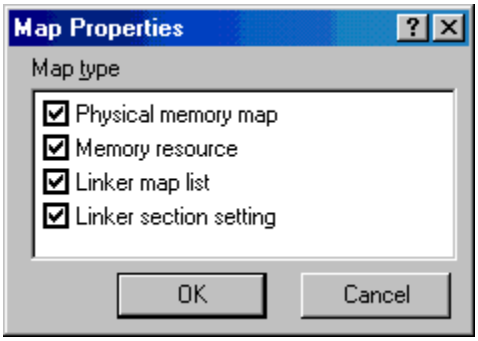
13. Map

The High-performance Embedded Workshop has a functionality to graphically display the map information of each map type. Map types that can be displayed depend on whether the toolchain is used or not, toolchain type and the debugger in use.

13.1 Graphically display map information of each map type

To graphically display map information of each map type:

1. Click the [Map] button () on the toolbar.
2. The [Map Properties] dialog box is displayed.



3. Select map types you wish to display from [Map type].

[Physical memory map]	The memory map will be displayed.
[Memory resource]	The memory resource will be displayed.
[Linker map list]	Use the toolchain and check [Generate list file] in the [List] category on [Link/Library]. The linker map list will be displayed after a build.
[Linker section setting]	Use the toolchain. Information set in the [Section] category on [Link/Library] will be displayed.

4. Click the [OK] button.

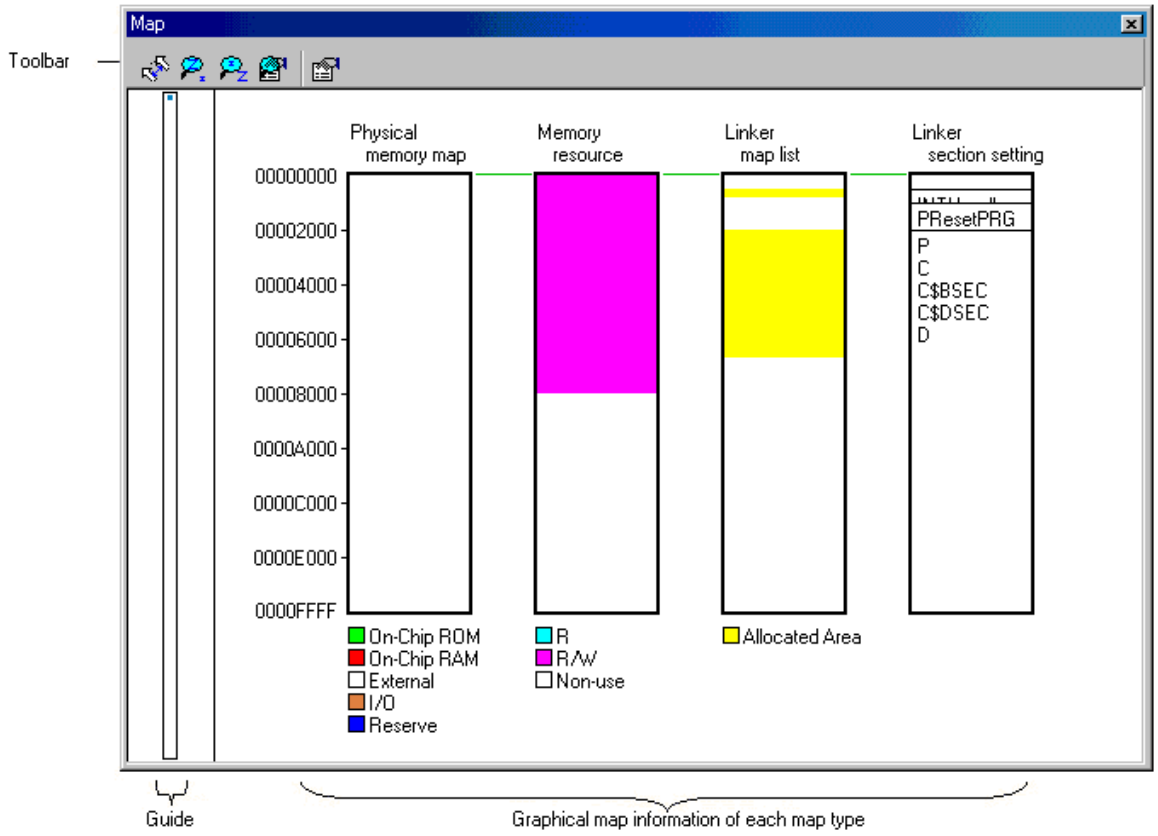
Note:

[Physical memory map] and [Memory resource] are displayed graphically when the simulator/debugger is connected and the memory map and memory resource are set. [Linker map list] and [Linker section setting] are displayed graphically when the toolchain is in use.

Configuration of Map Window

As shown in following figure, map information of checked map types is displayed graphically.

In following figure, a blue dot on the guide indicates the position of currently displayed addresses. The green line across the map types indicates the current cursor position.



- To display other addresses, Click on the guide.

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

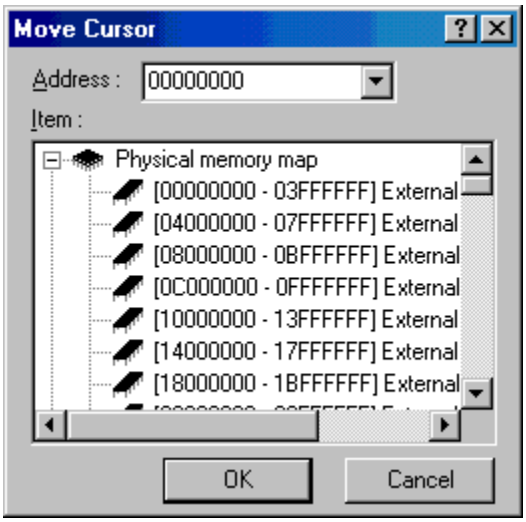
A basic operation is allocated to the toolbar.

The functions of [Toolbar display] and [Customize toolbar...] are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar bottom	Function
Move Cursor...		Changing the cursor position
Zoom In		Zooming in the display
Zoom Out		Setting back to the previous display
Zoom Properties...		Changing the zoom mode
Properties...		Changing the map type being displayed
Toolbar display	-	Showing/hiding toolbar buttons
Customize toolbar...	-	Customizing toolbar buttons

13.2 Changing the cursor position

1. Select [Move Cursor…] from the pop-up menu.
2. The [Move Cursor] dialog box is displayed.
3. Select an item for display. The first address of the selected item is displayed in [Address].
4. Click the [OK] button.



13.3 Zooming in the display

To zoom in the display:

1. Select [Zoom In] from the pop-up menu.

13.4 Setting back to the previous display

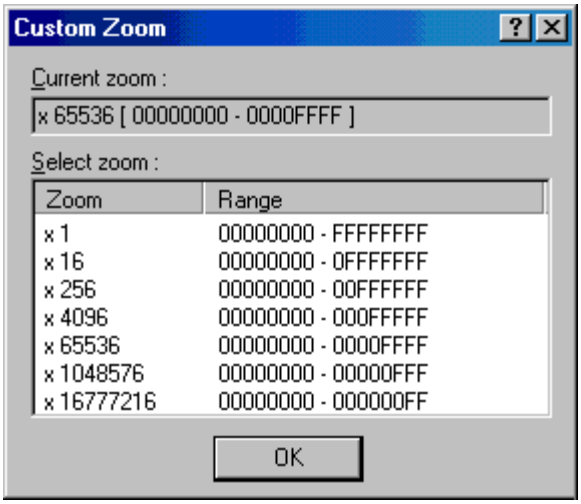
To set back to the previous display:

1. Select [Zoom Out] from the pop-up menu.

13.5 Changing the zoom mode

To change the zoom mode:

1. Select [Zoom Properties…] from the pop-up menu.
2. The [Custom Zoom] dialog box is displayed.
3. Select the zoom mode in [Select zoom].
4. Click the [OK] button.



13.6 Changing the map type being displayed

To change the map type being displayed:

1. Select [Properties…] from the pop-up menu.
2. The [Map Properties] dialog box is displayed.
3. Select map types you wish to display from [Map type].
4. Click the [OK] button.

14. Using the Command Line

The HEW Command Line Interpreter allows the user to control the debugging platform by sending text-based commands instead of the window menus and commands. It is especially useful if a series of predefined commands need to be sent to the debugging platform by calling them from a batch file and, optionally, recording the output in a log file.

Note:

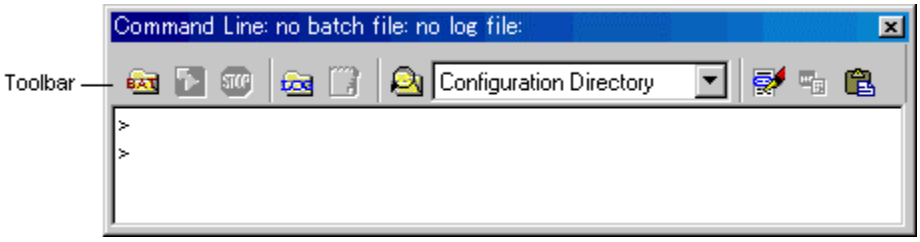
To specify a file in the command line, use a placeholder (excluding TCL). If you wish to specify a directory not included in the placeholder, specify an absolute path. After specifying the absolute path, this file will not be correctly found when it is in another host computer or environment where the path content is different. In such cases, specify the file again.

Example:FILE_LOAD ELF/DWARF2 $(CONFIGDIR)\\demo.abs

14.1 Opening the Command Line Window

Choose [View->Command Line] or click the [Command Line] toolbar button to open the [Command Line] window.

Configuration of Command Line window



- This window allows the user to control the debugging platform by sending text-based commands.
- A series of predefined command lines can be called from a file and the output can be recorded in a file.
- The command can be executed by pressing the Enter key after the command is input at the prompt (>) on the last line. For information about the available commands, refer to Reference 2, List of Commands, and the on-line help.
- If available, the window title displays the current batch and log file names separated by colons.
- Pressing the Ctrl + ↑ or Ctrl + ↓ keys on the last line displays the previously executed command line.
- The HEW command and TCL commands can be input in this window.

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

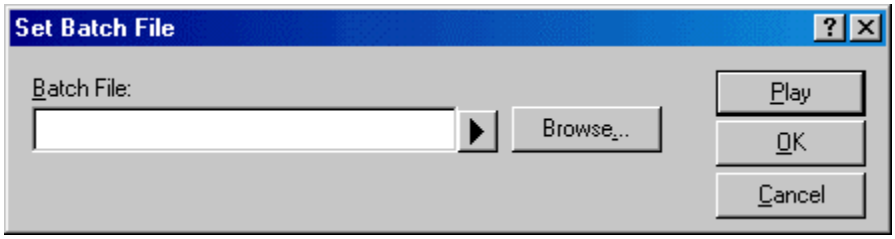
The functions of [Toolbar display] and [Customize toolbar...] are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options		Toolbar bottom	Function
Set Batch File...			Specifying a command file
Play			Executing a command file
Stop			Stopping command execution
Set Log File...			Specifying a log file
Logging			Starting or stopping logging
Browse...			Entering a full path to the file
Placeholder	Configuration directory	Configuration Directory	Pasting $(CONFIGDIR) placeholder
	Configuration name		Pasting $(CONFIGNAME) placeholder
	Project directory		Pasting $(PROJDIR) placeholder
	Project name		Pasting $(PROJECTNAME) placeholder
	Workspace directory		Pasting $(WORKSPDIR) placeholder
	Workspace name		Pasting $(WORKSPNAME) placeholder
	HEW Installation directory		Pasting $(HEWDIR) placeholder
Select All			Selects (i.e. highlights) the entire contents of the active window
Copy			Places a copy of the highlighted text into the Windows® clipboard
Paste			Copies the contents of the Windows® clipboard into the active window at the position of the insertion cursor
Toolbar display		-	Showing/hiding toolbar buttons
Customize toolbar...		-	Customizing toolbar buttons

14.2 Specifying a Command File

It is useful to use a command file when a series of predefined command lines need to be executed. Create a command file by a text editor and write necessary command lines. The default extension of a command file is .hdc.

Choose [Set Batch File...] from the pop-up menu to open the [Set Batch File] dialog box, in which the name of a command file (*.hdc) can be specified. Clicking the [OK] button displays the specified command file name as the window title. Clicking the [Cancel] button closes the dialog box without modifying the setting.



14.3 Executing a Command File

Click the [Play] button in the [Set Batch File] dialog box or choose [Play] from the pop-up menu to execute the command file. The [Play] menu is displayed in gray while the file is running and can be used when the command file execution stops and control returns to the user.

14.4 Stopping Command Execution

Choose [Stop] from the pop-up menu to stop command execution. The [Stop] menu becomes valid during command execution.

14.5 Specifying a Log File

Choose [Set Log File...] from the pop-up menu to open the [Open Log File] dialog box, in which a log file to store the command execution results can be specified.

Enter the name of a log file (*.log). The logging option is automatically set and the name of the file is shown on the window title bar.

Opening a previous log file will ask the user if they wish to append or overwrite the current log.

14.6 Starting or Stopping Logging

Choose [Logging] from the pop-up menu to toggle logging to file on and off. When logging is active, the button becomes effective. Note that the contents of the log file cannot be viewed until logging is completed, or temporarily disabled by clearing the check box. Re-enabling logging will append to the log file.

14.7 Entering a Full Path to the File

It is recommended that the full path to a file is specified as a file name in the [Command Line] window because the current directory can be moved. However, care must be taken to enter the correct full path to a file when it is entered from the keyboard. To save this trouble, a full path can be easily specified by browsing through files.

Choose [Browse...] from the pop-up menu to open the [Browse] dialog box. Select a file and click [Open] to paste the full path to the selected file to the cursor location. This option can only be used when the cursor is located on the last line.

14.8 Pasting a Placeholder

Select a placeholder from the [Placeholder] submenu in the pop-up menu to paste the selected placeholder to the cursor location. This function is only available when the cursor is located on the last line.

Placeholder sub-menu	Placeholder
Configuration directory	$(CONFIGDIR)
Configuration name	$(CONFIGNAME)
Project directory	$(PROJDIR)
Project name	$(PROJECTNAME)
Workspace directory	$(WORKSPDIR)
Workspace name	$(WORKSPNAME)
HEW Installation directory	$(HEWDIR)

14.9 Select All

Selects all contents output in the [Command Line] window.

14.10 Copy

Only available if a block of text is highlighted. This copies the highlighted text into the Windows® clipboard, allowing it to be pasted into other applications.

14.11 Paste

This option pastes the content of the Windows® clipboard to the current cursor location. This option can only be used when the cursor is at the last line.

15. Using the Debugger

The HEW Debugger is a Graphical User Interface designed to ease the development and debugging of applications written in C/C++ and assembly-language for Renesas microcomputers. Its aim is to provide a powerful yet intuitive way of accessing, observing and modifying the debugging platform in which the application is running.

Key Features

- Windows® GUI for debugging
- Intuitive interface
- On-line help
- Common 'Look & Feel'

Notes:

- For detailed information about debugging platform hardware, please refer to the separate Debugging Platform User Manual.

15.1 Preparations for debugging

This section describes the preparations for debugging your program. You will learn how to select and configure a debugging platform with which to debug, how to load the user program, and what the debugger sessions are.

15.1.1 Compiling for debug

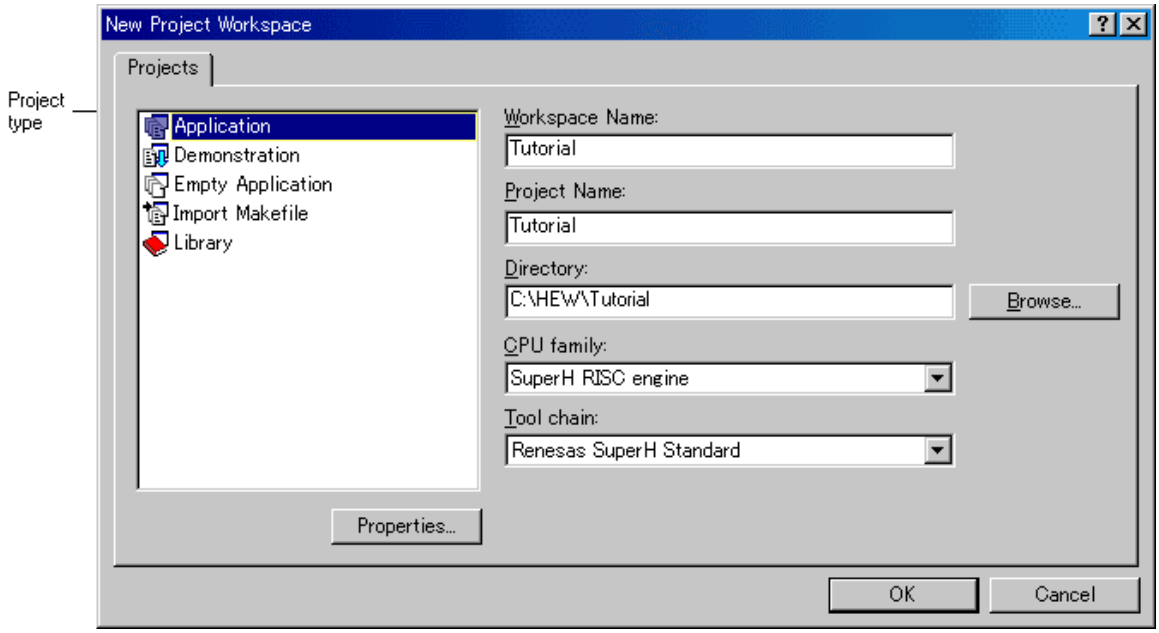
In order to be able to debug your program at C/C++ source level, your C/C++ program must be compiled and linked with the [Debug] option enabled. When this option is enabled, the compiler puts all the information necessary for debugging your C/C++ code into the absolute file or management information file, which are usually called **Debug Object Files**. When you create your project the initial setup will normally be configured for debug.

Notes:

- Make sure you have the debug option enabled on your compiler and linker when you generate an object file for debugging.
- If your debug object file does not contain any debugging information (for example, the S-Record format), then you can still load it into the debugging platform, but you will only be able to debug at the assembly language level.

15.1.2 Selecting a debugging platform

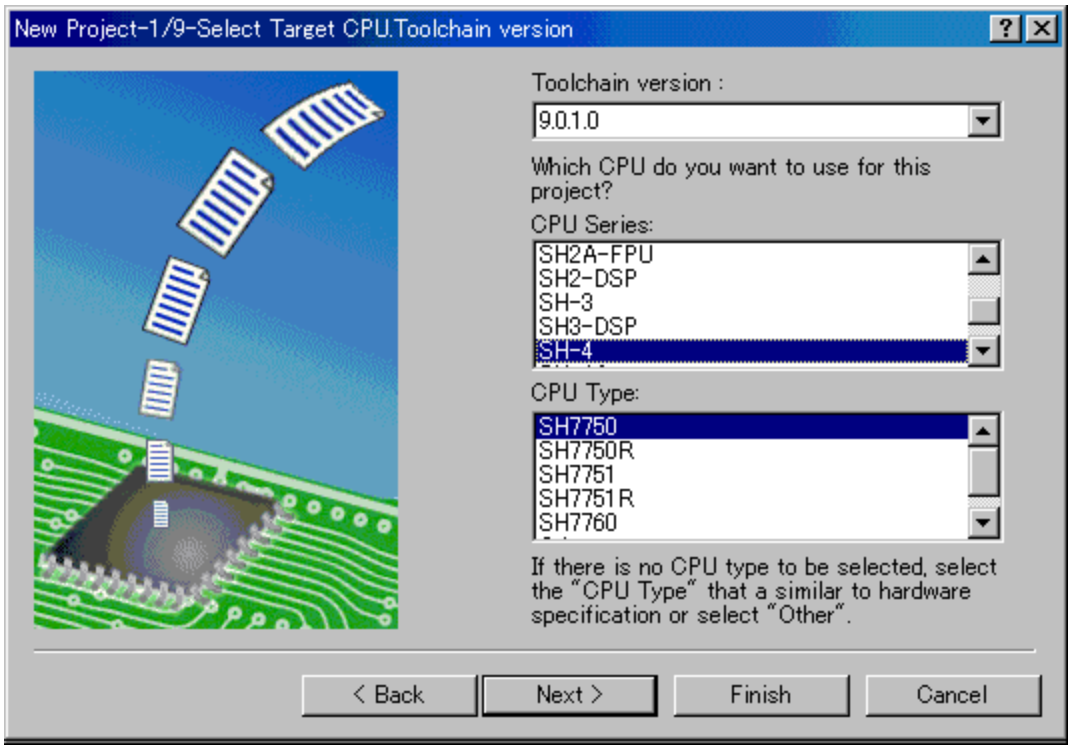
Selecting the debugging platform is very dependent on the installation of the HEW. If the HEW has a toolchain installed then the application project generator will be able to set up both the toolchain and the debugger targets simultaneously. This allows the options for targets and toolchain to be matched closely so that no inconsistencies occur. If there is no toolchain installed you will only be able to select debug-only project types. By default, HEW will display a debug-only project generation type for each CPU family in the **New Workspace** dialog. This project type will provide similar behaviour to the previous HEW session dialog box.



The **New Workspace** dialog allows you to select a project type for generation, which matches your CPU target.

Project Type	Description
[Application]	Project for generating an execution program that includes the initial routine file written in the C/C++ language.
[Assembly Application]	Project for generating an execution program that includes the initial routine file written in the assembly language.
[Demonstration]	Project for generating a demonstration program written in the C language.
[Empty Application]	Project for only setting the toolchain environment (no generation file).
[Import Makefile]	A project to create an executable program by importing an existing makefile.
[Library]	Project for generating a library file (no generation file).

In the example below, the SH-4 simulator/debugger is assumed. For details on the functions available with the debugging platform in use, refer to the user's manual or help of the debugging platform.

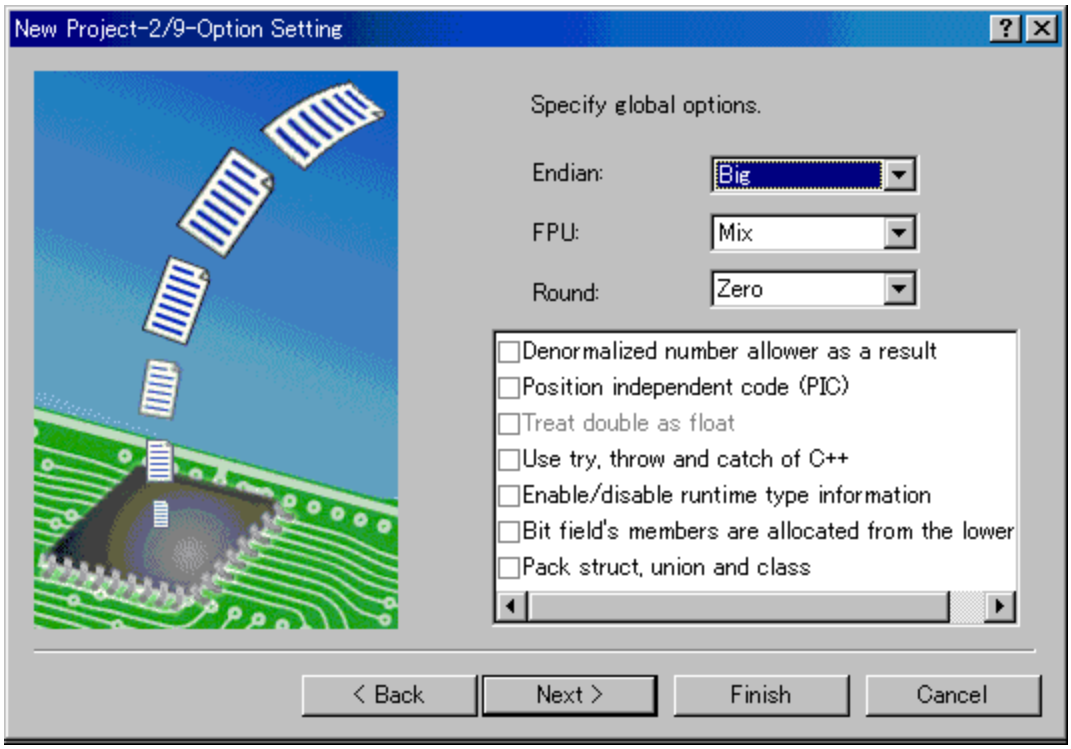


1. Select the CPU and Toolchain version in Step 1. The CPU types ([CPU Type]) are classified according to the CPU series ([CPU Series]). Select the CPU corresponding to the program to be developed because the generation file differs according to the [CPU Series] and [CPU Type] settings. If there is no corresponding CPU, select a CPU with similar hardware specifications or [Other].

The following buttons at the bottom of the dialog box are the same as those in the [New Project] wizard dialog box.

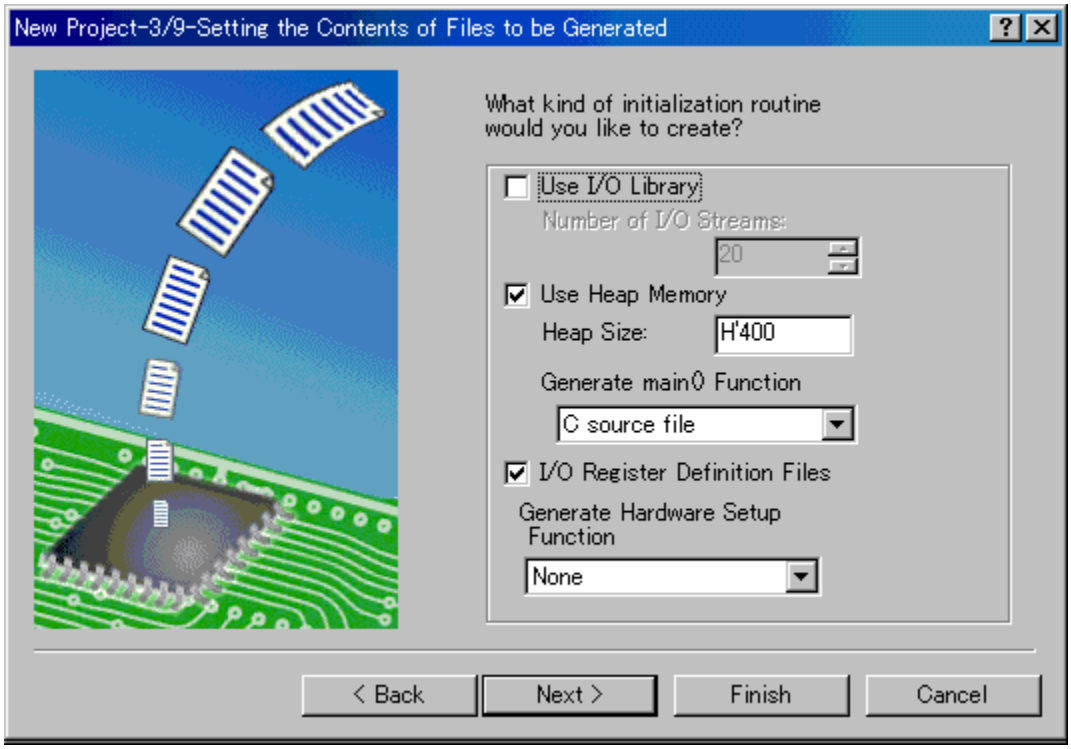
[Next>]	Moves to the next display.
[<Back]	Returns to the previous display.
[Finish]	Opens the [Summary] dialog box (selections followed by this button are default).
[Cancel]	Returns to the [New Project Workspace] dialog box.

To move to Step 2, click the [Next>] button in Step 1.



2. Specify the options common to all project files in Step 2. The specifiable items depend on the CPU selected in Step 1.

To move to Step 3, click the [Next>] button in Step 2.



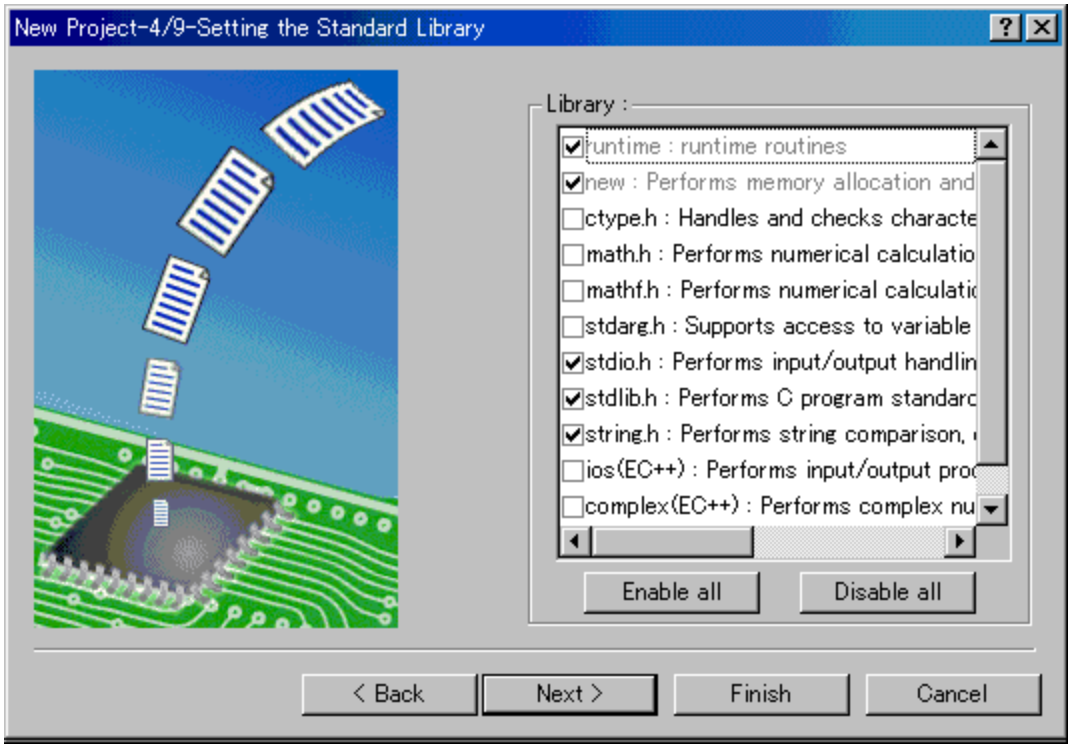
3. Specify the generation file in Step 3.

[Use I/O Library]	Checking enables use of standard I/O libraries.	
	[Number of I/O Streams]	Specifies the number of I/O streams that can be used simultaneously.
[Use Heap Memory]	Checking enables use of the heap area management function sbrk( ).	
	[Heap Size]	Specifies the unit of the size of the heap area to be managed.
	[Generate main( ) Function]	Selects generation of a model main function. Generates a main function file [(Project name).c(cpp)].
[I/O Register Definition Files]	Checking generates an I/O register definition file (iodefine.h) written in the C language.	
	[Generate Hardware Setup Function]	Selects generation of a model I/O register initial setting program. Generates a hardware setting file (hwsetup.c(cpp) or hwsetup.src).

Note:

To include a main function that has already been made, select [None] in [Generate main( ) Function] and after making the project, add the file containing the main function to the project. Note that if the name of the function to be included is different, the function calling section in resetprg.c must be modified. Be sure to refer to the hardware manual of the CPU for actual values of the sample file contents, such as the vector table definition and I/O register definition, which are generated by the project generator.

To move to Step 4, click the [Next>] button in Step 3.

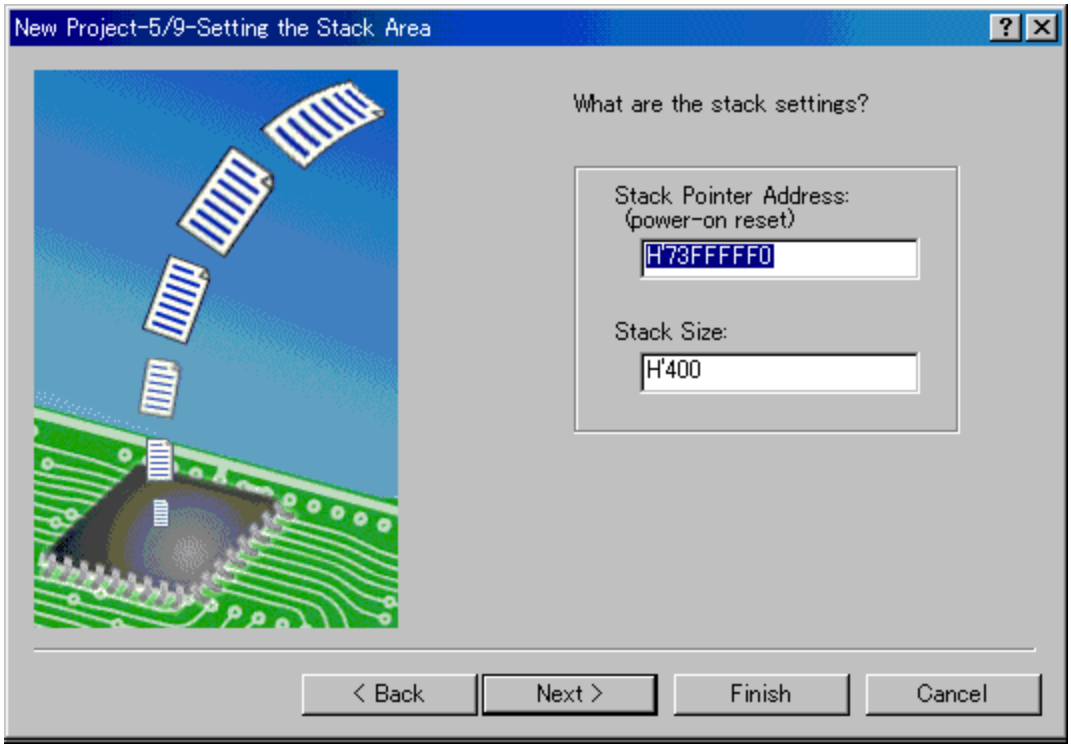


4. Specify the configuration of the standard libraries used by the C/C++ compiler in Step 4.

The functions defined in the checked items and the runtime function are included.

[Enable all]	Selects all standard library functions.
[Disable all]	Does not select all standard library functions. Note that only the minimum required functions, runtime and new, are selected.

To move to Step 5, click the [Next>] button in Step 4.

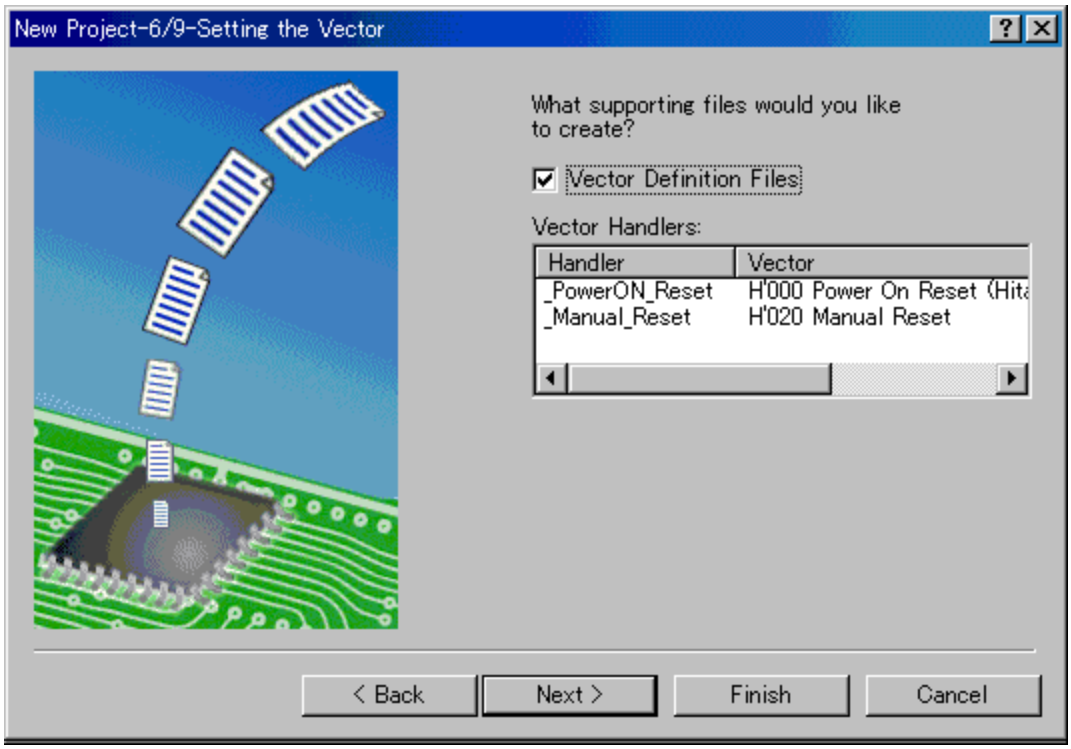


5. Specify the stack area in Step 5. This is done by setting the initial value of the stack pointer and the stack size. The initial value of the stack areas depends on the CPU selected in Step 1.

Note:

The stack area is defined by stacksct.h which is generated by the HEW. If stacksct.h has been modified by an editor, it cannot be modified from [Project -> Edit Project Configuration] in the HEW.

To move to Step 6, click the [Next>] button in Step 5.



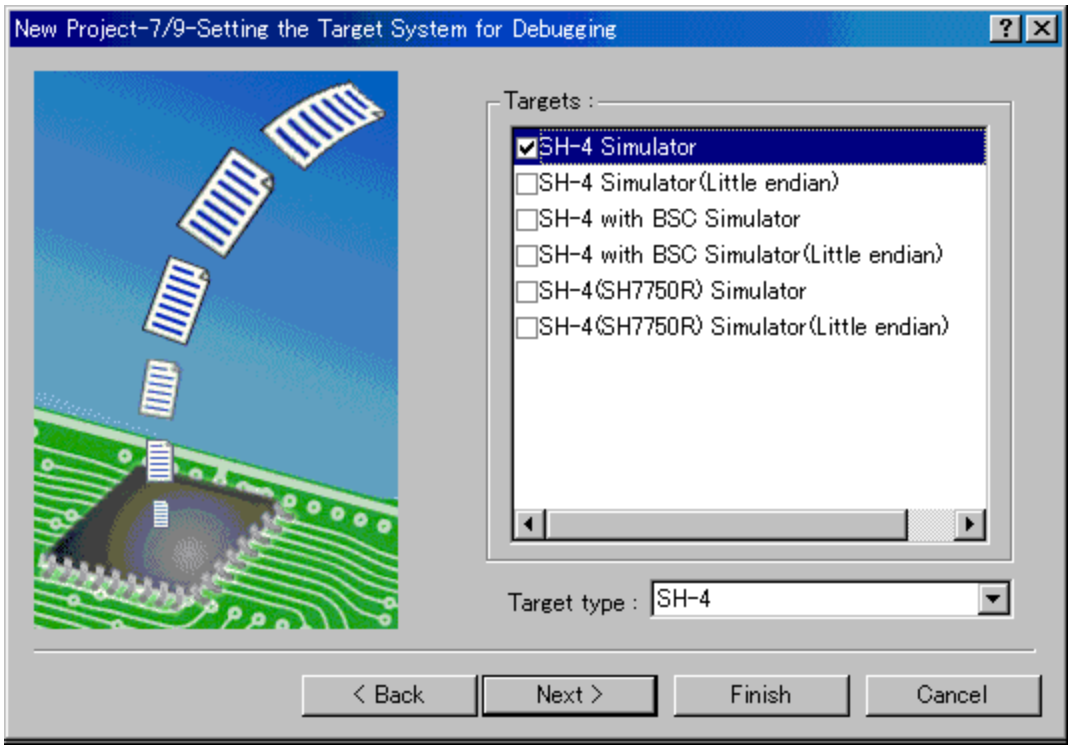
6. Specify the vector in Step 6.

[Vector Definition Files]	Checking generates a vector definition file and a vector table setting function definition file.	
[Vector Handlers]	[Handler]	Displays the handler program name of the reset vector. To modify the handler program, after selecting the handler program name by clicking on it, enter the new handler program name. Note that if the handler program is modified, a reset program (resetprg.c) is not generated.
	[Vector]	Displays a description of the vector.

Note:

Since the generated reset program, interrupt functions, reset vector handlers, and interrupt source register definitions are samples, be sure to refer to the CPU hardware manual.

To move to Step 7, click the [Next>] button in Step 6.



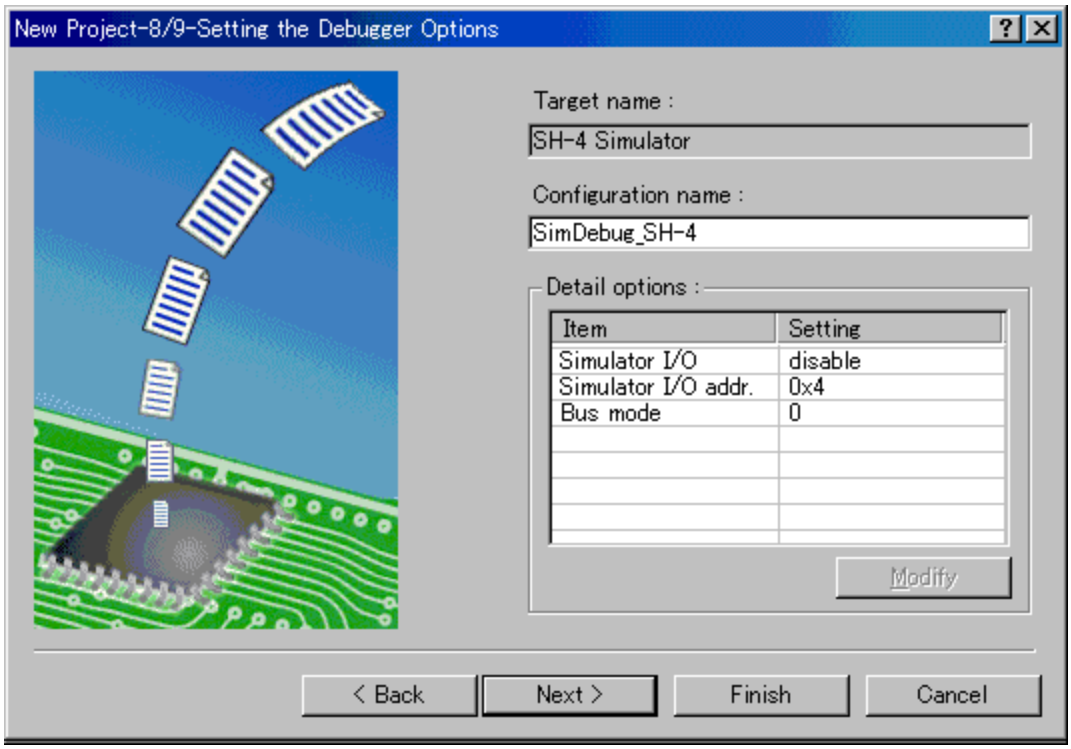
7. Specify the debugger targets in Step 7.

[Targets]	Sets the debugger targets. Select (by checking) the debugger targets. No selection or a selection of more than one target is possible.
[Target Type]	Specifies the type of the targets displayed in [Targets].

Note:

The endian type selected in step 2 will be applied to the compiler settings. This is separate from the endian type of the debugger target selected in step 7.

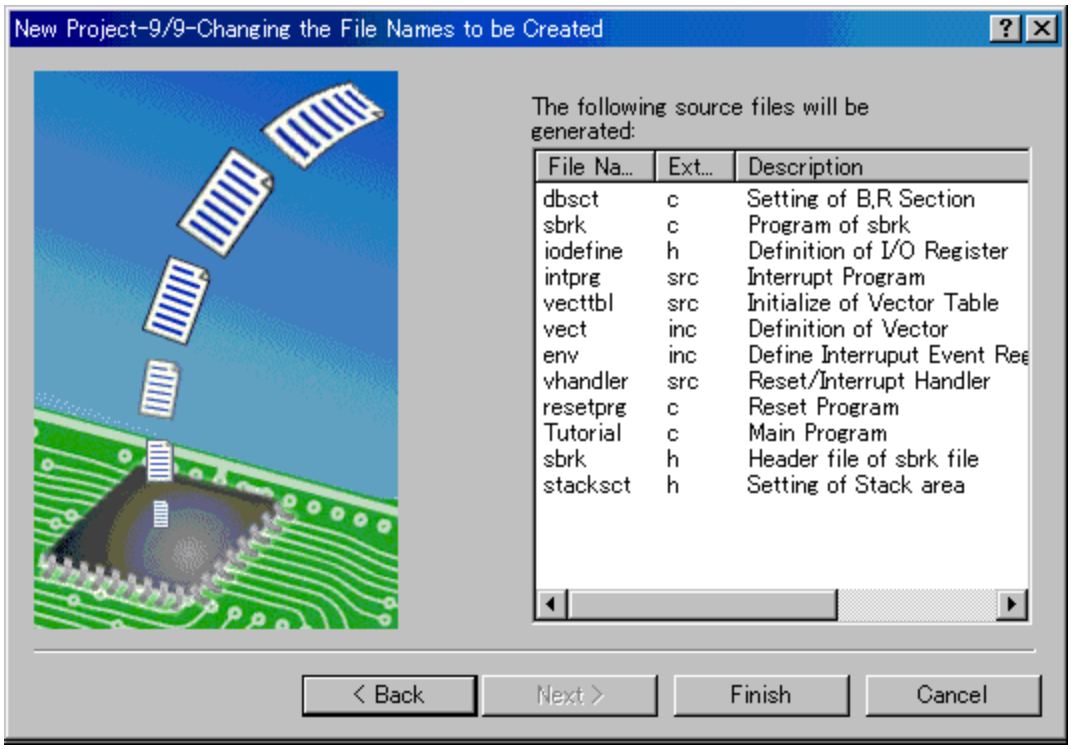
To move to Step 8, click the [Next>] button in Step 7.



8. Set the options for the debugger targets selected in Step 8.

[Configuration name]	By default, the HEW generates two configurations: [Release] and [Debug]. If a debugger target is selected, a configuration for the selected target is also generated (an abbreviation including the target name). This configuration name can be changed in [Configuration name].
[Detail options]	Sets the debugger target options. To modify an option, select [Item] and click [Modify]. If the selected item cannot be modified, [Modify] remains gray even when [Item] is selected.
[Simulator I/O]	System call for standard I/O or file I/O from the user program is enabled ([Enable]) or disabled ([Disable]).
[Simulator I/O addr]	Address for above system call.
[Bus mode]	Currently cannot be used by the simulator/debugger.

To move to Step 9, click the [Next>] button in Step 8.



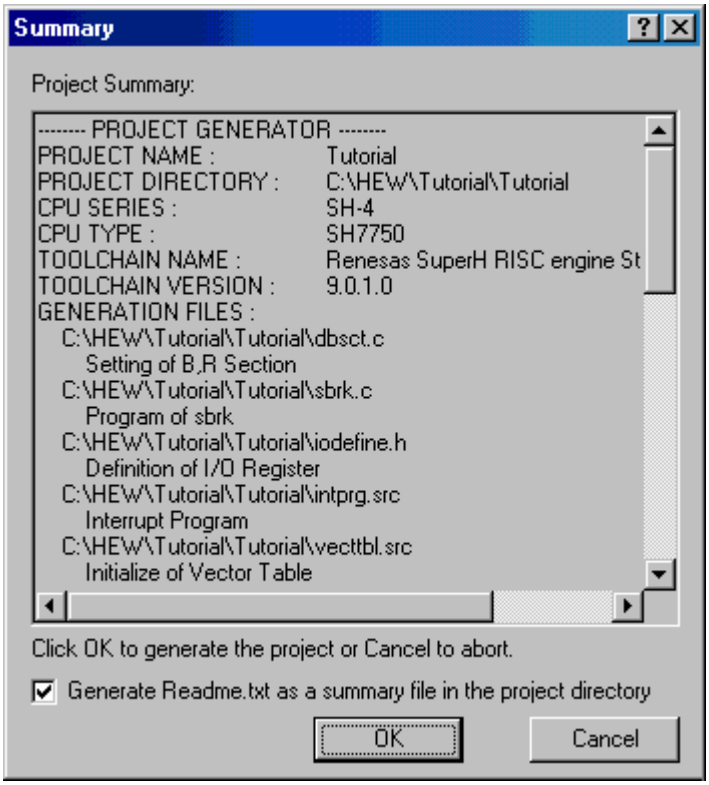
9. The files to be generated by the HEW based on the settings made so far is displayed as a list in Step 9.

[File Name] File name
To change a file name, after selecting the file name by clicking on it, enter the new file name.

[Extension] File extension

[Description] Description of the file

Clicking the [Finish] button in Step 9 displays the [Summary] dialog box.

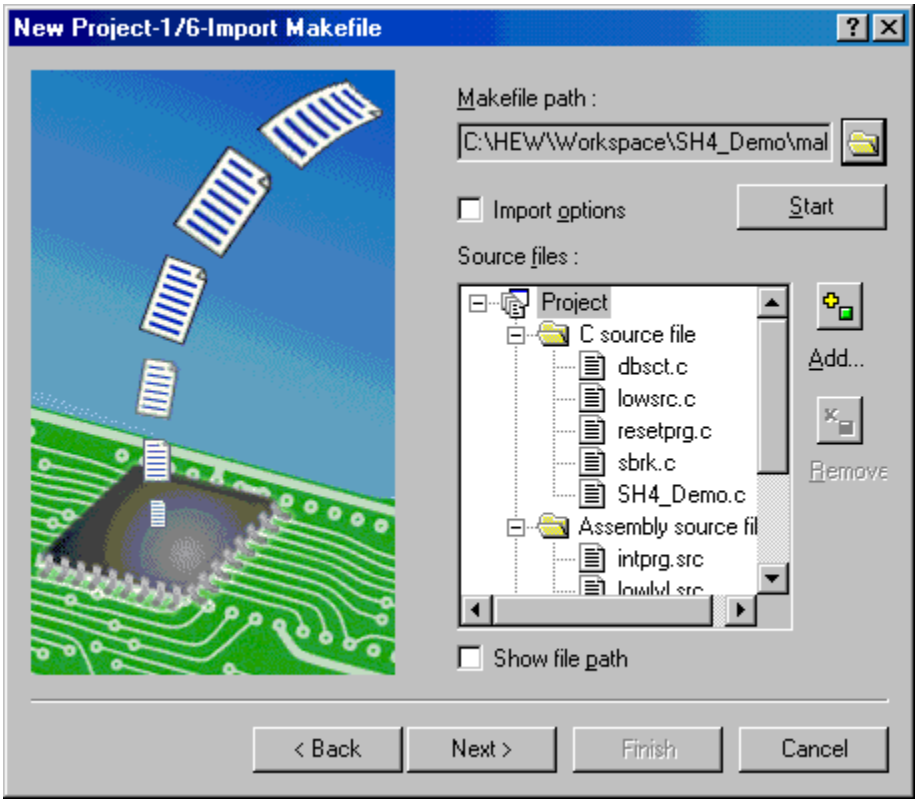


10. The project generator displays information on the project to be generated in the [Summary] dialog box. After confirming the display contents, click the [OK] button. Clicking [Cancel] returns to the [New Project] wizard dialog box. Checking [Generate Readme.txt as a summary file in the project directory] will save the project information displayed in the [Summary] dialog box as a text file named Readme.txt in the project directory.

(1) To Create a New Project Having Information from Makefile

HEW can analyze GNU make format and Hmake format (HEW generated) and create a workspace which has file information from makefile.

Open the [New Project Workspace] dialog box and select the [Import Makefile] as project type. After supplying some fields (e.g.: Workspace Name) and pressing [OK] button, [New Project – Import Makefile] dialog box is appeared.



When a makefile is selected by [Makefile path], [Source files] shows source files in the makefile. To view the source files in the makefile again, click [Start].

To apply toolchain options such as the compiler, select the [Import options] check box.

If you want to remove a file from the project, you can remove it by selecting the file and pressing [Remove] button. And if you want to add a file to the project, you can add it by pressing [Add] button.

Selecting the [Show file path] check box shows the full path of the file.

15.1.3 Editing Project Configuration

If you are using the H8 or SH toolchains then it is possible to configure the simulator again using the project generator. This feature is not enabled for the demonstration project type.

1. Click on the [Project->Edit Project Configuration...]. The [Edit Project Configuration] dialog is displayed.
2. Click on the [Target] tab.
3. Select the target you wish to use and then click [OK].

15.1.4 Configuring the debugging platform

Before you can load a program into your debugging platform you must set it up to match your application's system. The items that must be set-up are typically device type, operating mode, clock speed and the memory map. It is particularly important to set-up the memory map, as you must have memory in the debugging platform, into which your user code will be loaded.

In the High-performance Embedded Workshop, the project generation process will have completed much of this work. However if you are using a different configuration of board from the standard types then some customization will be essential.

(1) Setup

To set-up the debugging platform configuration choose the [Setup→Simulator] menu option or the [Setup→Emulator] menu option. Under this sub menu will be the menus which can be used to configure your debug platform.

In the case of the SH family Simulator the available menus are the [System...] and [Memory resource...]. These options both allow the simulator to be customized and setup to your requirements.

You will be presented with a set-up dialog specific to the debugging platform that you chose in the [Debug Settings] dialog.

Note:

For a detailed description of the features available in your debugging platform, please refer to the separate Debugging Platform User Manual.

(2) Memory mapping

For the debugger to correctly represent your user system, the memory map must be set up. It needs to know which areas in the device's address space are RAM, ROM, on-chip registers or areas where there is no memory.

When you select the device type and mode in the project generator, the HEW will automatically set up the map for that device and the mode in which the processor is operating. For example, in a device with internal ROM and RAM, the areas where these are located in the device's memory map will be set by default.

If you are using a device that does not have internal memory, or a device with external memory instead of (or in addition to) the internal memory, then you must tell the debugging platform that you have memory there.

Tip:

If you are trying to debug code with an emulator and need some memory available that does not exist either on-chip or externally (in your hardware), then you can map some *emulation memory* from the emulator to the address space for your application to use.

The details will be specific to the debugging platform that you chose in the new project.

Additional information about the memory mapping can be viewed in the [System Status] view's [Memory] pane. The [Device Configuration] area shows how the memory in the device's address space.

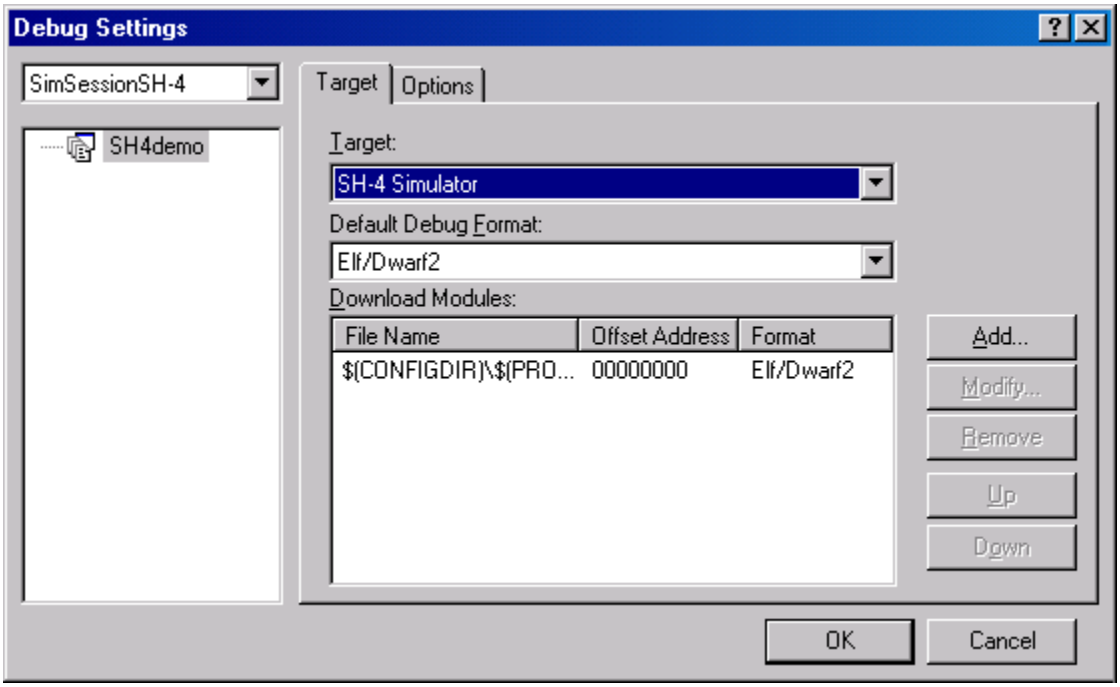
Note:

Due to page length limitations in some emulators, the range addresses may not exactly match the entered addresses.

(3) Setting up the HEW debugger

Normal operation of the HEW debugger means that your target and download modules will be automatically configured in the project generation process. However in some cases it may be necessary for you to manually configure your debug session. This is often the case when using old toolchains and project generators that do not support the latest HEW interfaces.

To check your debug session setup click on the [Debug->debug Setting...] menu item. The [Debug Settings] dialog is displayed:



From this dialog it is possible to choose the target, default debug format and download modules. The Options tab also gives access to command line batch files and connection and download module options.

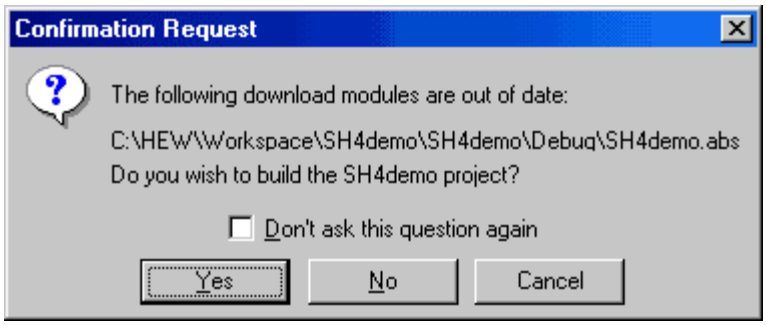
To change the target the following operations is necessary:

1. Select the project that needs to be changed in the tree on the left of the dialog. It defaults to the current project.
2. Select the session which is to be modified in the drop list above the tree.
3. Change the target using the target drop list control. This removes any target specific setup options that have been previously been set.

(4) Check for changed source files before download

The HEW can check to see if any of the source files have changed before the download module is downloaded in the current project.

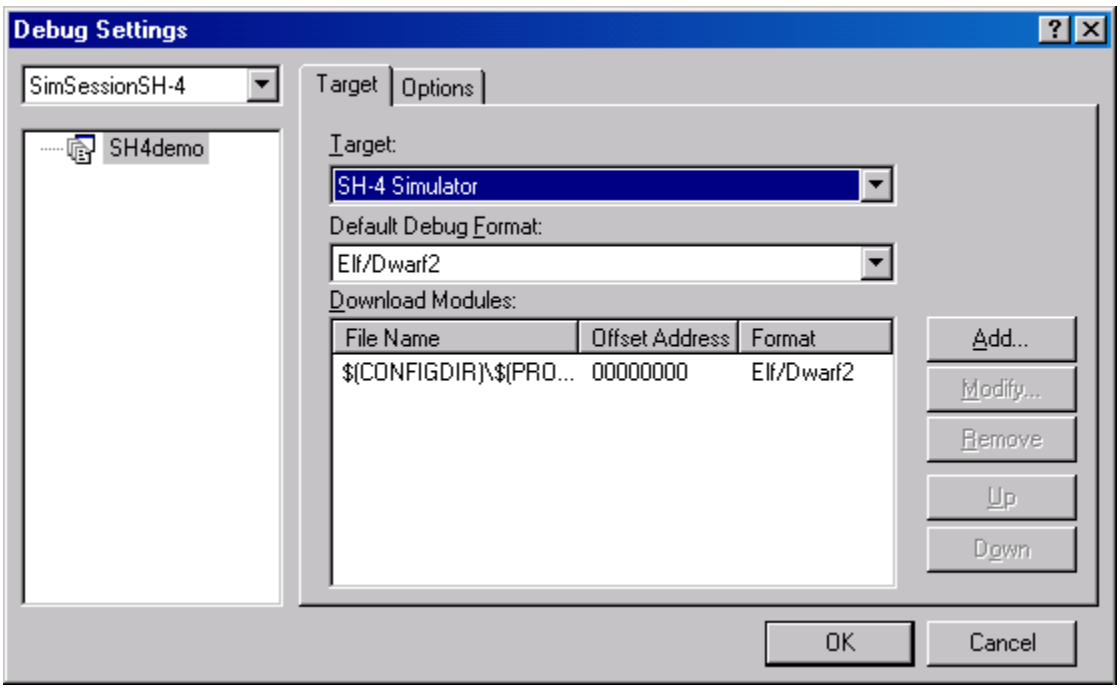
If files have been modified then a confirmation is launched which asks the user if they wish to rebuild the code before the download takes place.



This is a confirmation dialog so can be switched of if this feature is not desirable. This is available from the [Setup->Options...] dialog on the [Confirmation] tab. This is shown below.

(5) Setting the Downloading a Program

Once you have made sure that there is memory in your system in which to download your code, you can then proceed to download a program to debug. The initial selection of download module is automatic with regard to an application generator, as it is the output from the linker.



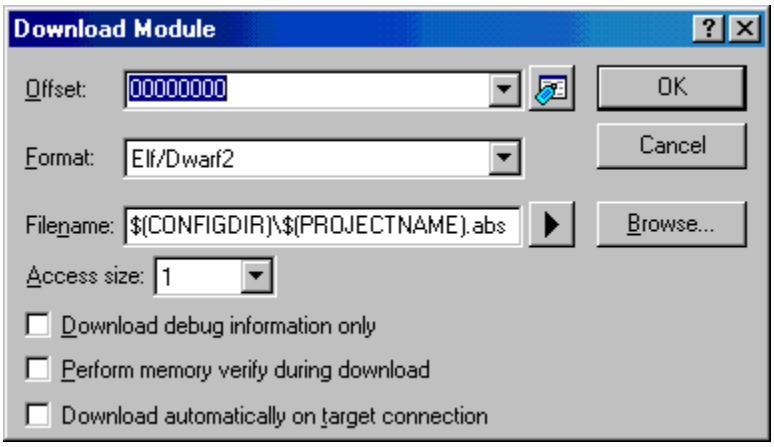
It is also possible to manually choose download modules after the project creation. This is achieved via the [Debug Settings] dialog. This dialog allows you to control the debug settings throughout your workspace. The tree on the left of the dialog contains all of the current projects. Selecting a project in this tree will then show you the settings for that project and the configuration selection in the [Configuration] drop-down list. In this list box, it is possible to select multiple configurations or all configurations. If you select multiple configurations you can choose to modify the settings for one or more configurations at once. The [Debug Settings] dialog displays the following debug options:

- Current debug target for the current project and configuration selection.
- Download modules for the current project and configuration selection.

The download module list displays the order in which the files will be downloaded to the target. It is possible to Add, Remove, Modify, Up and Down modules in this list.

To add a new download module:

1. Select the [Debug→Debug Settings...] menu option. The [Debug Settings] dialog will be displayed.
2. In the project tree, select the project and configurations to which you want to add a download module.
3. Click the [Add...] button. The [Download Module] dialog is displayed.



4. All fields must be setup for the download module to be configured correctly. The first field is [Offset]. This is the memory address offset the module will be loaded at. It defaults to 0.
5. The [Format] drop-down list box contains a list of supported object format. Note this does not have to match the default object format. However you can only debug modules that match the format specified in the debug settings dialog default object format field.
6. The [Filenames] can be specified with placeholders or as an absolute setting. It is recommended to use placeholders in the same manner as $(CONFIGDIR)\$(PROJECTNAME).abs.
7. The [Access size] field specifies the access width when the memory is accessed. This option is often used by hardware design engineers when testing their hardware. For the purposes of this demonstration the default value should be used.
8. The [Download debug information only] check box can be used when we cannot download the actual module to the memory on the target. For example the program may be resident in a read-only memory type like flash. In this instance you can download just the debug information to RAM so that it is still possible to debug the user code.
9. The [Perform memory verify during download] checkbox can be used to do additional checks when downloading the module to ensure it was correctly downloaded to the target device.
10. The [Download automatically on target connection] checkbox can be used to automatically download the module when the target is being connected.
11. When you click the [OK] button, the debug download module is added to the bottom of the list. If you wish to position the module in a different position in relation to the other modules, select the module and then use the [Up] and [Down] buttons to position the module correctly.

Any changes made in the [Debug Settings] dialog are only changed when you click OK.

The default debug format is set to the first download module in the list by default. Only one default debug object format can be specified for each session. All currently installed debugger formats are listed here.

(6) Manually downloading modules

Once you have decided which download modules are to be downloaded to the target, it is possible to manually update the modules on the connected target. This is achieved by selecting the [Debug->Download Modules] menu item. This is a cascading menu item which allows a single module or all of the modules to be downloaded.

This can also be achieved by right-clicking on modules under the [Download Modules] folder in the [Workspace] window.

(7) Unloading of Modules

It is possible to manually unload downloaded modules in the list whenever you want from the [Debug->Unload Modules] menu item. This can also be done from the [Download Module] pop-up menu in the [Workspace] window.

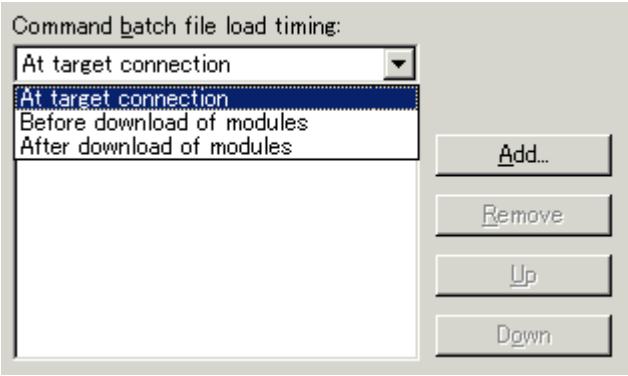
When a module is unloaded, its symbols are erased from the HEW debugging system, but the memory contents of the target remains unmodified. After a module has been unloaded, it cannot be debugged unless it is reloaded.

(8) Setting the Option of Debug Settings

The HEW debugger is tightly integrated with the TCL command line facilities. This means that it is possible to write batch files for the HEW debugger which can be executed automatically at certain times.

To configure the automatic command line batch file execution:

1. Select the [Debug->Debug Settings...] menu item.
2. Select the [Options] page.
3. Select the [Command batch file load timing]. This can be the following values, at target connection, before downloading the modules and after downloading the modules.



4. Then click [Add...]. The debugger will then display the add [Command Line File] dialog.
5. You can then specify the location of the file, by either browsing to the file or using placeholders.
6. Clicking the [Apply to all timings] check box adds the batch file to every timing.
7. Click [OK] to add the batch file.
8. Once added it can moved into the correct place in the order by using the [Up] and [Down] buttons. This is only valid if you are running multiple command line batch files.

Other Options:

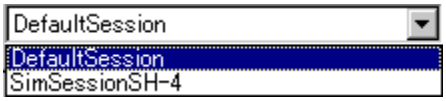
[Do not perform automatic target connection] *1	If the option is checked, the target is not connected until you select [Debug->Connect]. If the option is unchecked, the target is connected whenever a configuration is opened. This happens when a new workspace or project is opened or also when you switch configuration using the toolbar or [Build Configurations] dialog box.
[Download modules after build]	If the option is checked, the user program will be automatically downloaded after a build.
[Reset CPU after download module]	If the option is checked, the debugger target will be automatically reset after downloading the user program.
[Remove breakpoints on download]	If the option is checked, the breakpoints will be automatically removed after downloading the user program.

***1 :** Support for this function depends on the debugging platform.

15.1.5 Debugger sessions

The HEW allows you to store all of your builder options into a configuration. This means that you can "freeze" all of the options and give them a name. In a similar way, HEW allows the user to store his debugger options in a session. Later on, you can select the session and all of the debugger options will be restored. These sessions allow the user to specify target, download modules and debug options. This means that potentially each session can be targeted at a different end platform.

This facility can allow you to have many different sessions, each with different debugger options defined. For example, it is possible to have each session using the same target but with slight variations in the session options. This can mean it is very easy for the user to switch session and modify such things as register values, or target settings such as clock speed. The figure below shows this principle. The two sessions share the same target but the sessions can be slightly different, with regard to the options defined. This means that both sessions can share the same download module and avoid an unnecessary rebuild. This is because sessions are not directly related to the build configuration data.



Each session's data is stored in a separate file to the HEW project. You can then manipulate the data to share or modify as is required in the project.

(1) Selecting a session

To select a session, select one of the following operations:

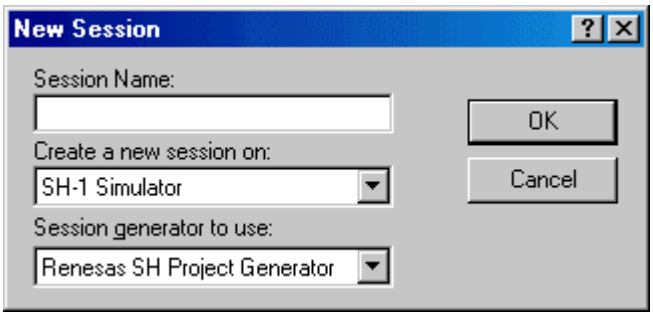
- Select it from the drop down list-box in the toolbar, OR
- Choose the [Debug→Debug Sessions…] menu option. The [Debug Sessions] dialog box will be displayed. Select the session that you want to use from the [Current session] drop down list-box and click the [OK] button.

(2) Adding a session

You can now create a session with a target attached and setup ready for use. This session can be given a name and the target chosen.

To create a new session, with a target attached and setup:

1. Choose the [File→New Session…] menu option. The [New Session] dialog box is displayed.



2. Enter the new session name.
3. Select the target you wish to use in the new session.
4. Choose the session generator to use. This should default to the correct selection. However sometimes there may be multiple generators that support the same target.
5. Click the [OK] button. This should launch the generation process, the process depends on the session generator that was selected. At this point an additional dialog may be displayed for target setup options.
6. When finished, a new session is added to the current project. It should be available in the sessions drop list box on the main toolbar.

You can create a new empty session in the project directory. The session will use the session name as its new file name. If the file name already exists then an error is displayed.

To add a new empty session:

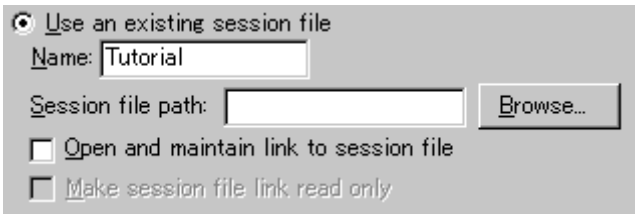
1. Choose the [Debug→Debug Sessions…] menu option. The [Debug Sessions] dialog box will be displayed.
2. Click the [Add…] button. The [Add Session] dialog box will be displayed.
3. Click the [Add New Session] radio button.
4. Enter a name for the session.

5. Click the [OK] button to close the [Debug Sessions] dialog box.

You can import session data from another file and create a new session file in the project directory. All information is an exact copy of the file the data was imported from.

To import an existing session into a new session file:

1. Choose the [Debug→Debug Sessions…] menu option. The Debug Sessions dialog box will be displayed.
2. Click the [Add…] button. The [Add Session] dialog box will be displayed.
3. Click the [Add An Existing Session File] radio button.
4. Enter a name for the session.
5. Browse to an existing session file location, which you would like to import into the current project.
6. Click the [OK] button to close the [Debug Sessions] dialog box.



This operation can also be achieved by using the [File->Import session...]

To import an existing session using [File->Import Session...]:

1. Choose the [File→Import Session…] menu option. The [Session Name] dialog box is displayed.

2. Enter the new session name.
3. Select the session file you wish to import into the new session.
4. Click [OK]. A new session as added with the same settings as the file you browsed to but with the new name.

(3) Importing a link to a session

You can add a new session to the HEW system but link to the session file in its location rather than importing or copying the file to the project directory. This is useful when sharing debugger information with other users in a network environment.

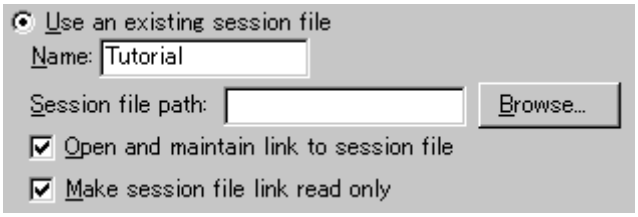
To import a link to an existing session file:

1. Choose the [Debug→Debug Sessions…] menu option. The [Debug Sessions] dialog box will be displayed.
2. Click the [Add…] button. The [Add new session] dialog box will be displayed.
3. Click the [Use an existing session file] radio button.
4. Enter a name for the session.
5. Browse to an existing session file location, which you would like to import into the current project.
6. Click the [Open and maintain link to session file] checkbox. This means the session will not be imported into the project directory but instead the HEW will link to the session location. This file location was entered in step 5 and it will save all of the session data in this location.
7. Click the [OK] button to close the [Debug Sessions] dialog box.

It is possible to make the link to session file read-only. This is useful if you are sharing debugger-setting files and you do not want data to be modified accidentally.

To import a link to an existing session file and make it read-only:

1. Choose the [Debug→Debug Sessions…] menu option. The [Debug Sessions] dialog box will be displayed.
2. Click the [Add…] button. The [Add new session] dialog box will be displayed.
3. Click the [Use an existing session file] radio button.
4. Enter a name for the session.
5. Browse to an existing session file location, which you would like to import into the current project.
6. Click the [Open and maintain link to session file] checkbox.
7. Click the [Make session file link read only] checkbox. This means that the HEW will be unable to save changes to this session and will only be able to read the data when the session is opened.



(4) Removing a session

To remove a session:

1. Choose the [Debug→Debug Sessions…] menu option. The [Debug Sessions] dialog box will be displayed.
2. Select the session you would like to remove.
3. Click the [Remove] button.
4. Click the [OK] button to close the [Debug Sessions] dialog box.

Note:

It is not possible to remove the current session.

(5) Making a session read-only

To make a session read only:

1. Choose the [Debug→Debug Sessions…] menu option. The [Debug Sessions] dialog box will be displayed.
2. Select the session you would like to view the properties for.
3. Click the [Properties...] button. The properties dialog is displayed.
4. Click the [Read only] checkbox. This makes the link read only. This is useful if you are sharing debugger-setting files and you do not want data to be modified accidentally.
5. Click the [OK] button to close the [Debug Sessions] dialog box.

(6) Saving Session Information

To save a session:

1. Select the [File] menu from the menu bar. The file menu is displayed.
2. Click the [Save Session] menu item.

If you have the [Prompt before saving session] checkbox checked, a dialog is displayed which asks you whether you wish to save the information. Clicking [No] loses the changes you made in the session. This checkbox is located in the [Setup->Options] dialog on the [Workspace] tab.

To save a session with a different name:

1. Choose the [Debug→Debug Sessions…] menu option. The [Debug Sessions] dialog box will be displayed.
2. Select the session you would like to save.
3. Click the [Save as...] button. The [Save Session] dialog is displayed.
4. Browse to the new file location.
5. If you only want to export the session file to another location then leave the [Maintain link] checkbox unchecked. If you would like HEW to use this location instead of the current session location then check the [Maintain link] checkbox.
6. Click the [OK] button.

To save a session with a different name using [File->Save Sessions As...]:

1. Choose the [File->Save Sessions As…] menu option. The [Session Name] dialog box is displayed.

2. Enter the new session name.
3. Click the [OK] button.

(7) Reloading Session Information

To reload a session:

Select [File->Refresh session]. Clicking this will lose any changes to your session currently and the reload the current session into HEW. A confirmation will be displayed before the reload happens.

(8) Debugging Multiple Targets

For the method to debug multiple targets in synchronization, refer to section 15.17, Synchronizing Multiple Debugging Platforms.

15.2 Viewing a program

This section describes how to look at your program as source code and assembly language mnemonics. The HEW has various facilities for dealing with code and symbol information which are explained in this section and you will be shown how to look at text files in the user interface.

Note:

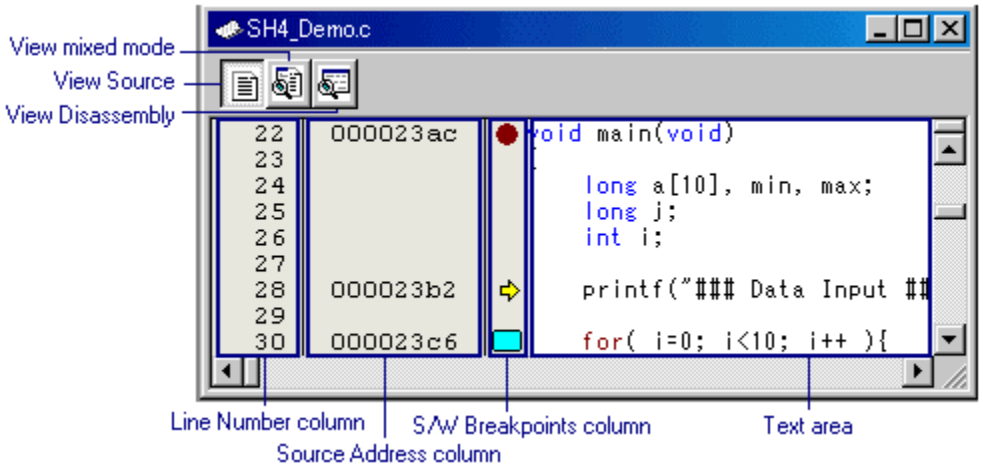
After a break occurs, the HEW displays the location of the program counter (PC). In some cases, for example if a project has been moved from its original path, the source files may not be automatically found. In this case the HEW will open a source file browser dialog to allow you to manually locate the file – this path will then be used to update any other source files in this debug project.

15.2.1 Viewing the code

To view a source file's code, double-click on its icon in the file tree, or right-click on the source file and click the [Open] option on the pop-up menu. The HEW opens the file in the integrated editor.

The editor is divided into two areas: the **Gutter** area (containing markers for breakpoints, PC location, etc.) and the **Text** area (containing color syntax highlighted code). This is shown in the figure below.

Configuration of Editor window



Line Number column	Displays the line number for the source file.
Source Address column	After your program has been downloaded, the [Editor] window displays the addresses for the current source file.
S/W Breakpoints column	Display the PC location ( ), breakpoints ( ) and bookmark ( ). Setting PC breakpoint by double-click.
Text area	Containing color syntax highlighted code

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

Pop-up menu options		Function
Build File		Build the selected files.
Cut		Removes highlighted text and places it on the Windows® clipboard.
Copy		Places a copy of the highlighted text into the Windows® clipboard.
Paste		Copies the contents of the Windows® clipboard into the active window at the position of the insertion cursor.
Add File To Project		Add file to a project.
Find...		Find text in the current file.
Find in Files...		Find text in multiple files.
Replace...		Replace text in the current file.
Goto Line...		Jump to a line in a file.
Match Braces		Find a matching brace.
Bookmarks	Toggle Bookmark	Sets a bookmark at the current line or clears a bookmark at the current line.
	Next Bookmark	Jumps to the next bookmark in the current file from the current line.
	Previous Bookmark	Jumps to the previous bookmark in the current file from the current line.
	Clear All Bookmarks	Clears all bookmarks in the current file.
Templates	Define Templates...	Defines a template.
	Insert Template...	Inserts a template.
Toggle Breakpoint		Sets or clears a software breakpoint at the line showing the address.
Enable/Disable Breakpoint		Enable or disable the current software breakpoint.
Define Column Format...		Set the status of editor columns.
Columns	Column name	Set the status of editor columns.
Instant Watch ***1**		Launches the Instant Watch dialog box with the name extracted from the view at the current text cursor (not mouse cursor) position.
Go To Cursor		Starts executing the user program at the current PC and continues until the PC equals the address indicated by the current text cursor (not mouse cursor) position.
Set PC Here		Changes the value of the Program Counter (PC) to the address at the row of the text cursor (not mouse cursor).
Display PC		Opens the [Editor] or [Disassembly] window at the address of the PC.
View Disassembly		Opens a Disassembly view at the address mating the current source line.

***1 :** Support for this function depends on the debugging platform.

15.2.2 Viewing assembly-language code

If you have a source file open, right-click to open the pop-up menu and select the **View Disassembly** option to open a **Disassembly** view at the same address as the current **Source** view.

It is also possible to view the disassembly using the new integrated disassembly view in the source file.

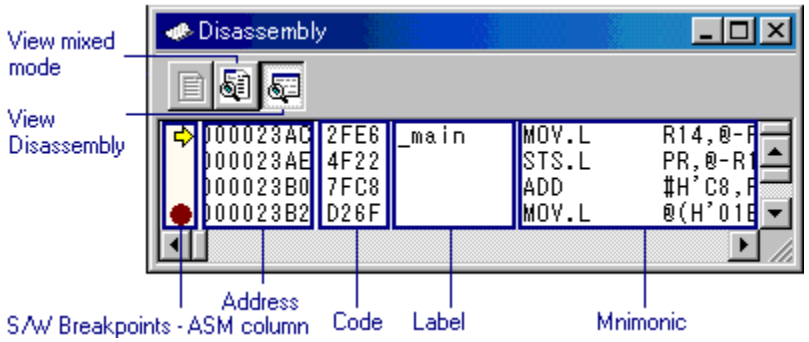
If you do not have a source file, but wish to view code at assembly-language level, then select one of the following operations:

- Click on the View Disassembly toolbar button , **OR**
- Choose the [View→Disassembly] menu option, **OR**
- Press CTRL+D.

The **Disassembly** view opens at the current PC location ( ), and shows **Address**, **Code** (optional) and **Assembler**, which shows the disassembled mnemonics (with labels when available). Optionally, any source line starting at that address may be shown, thus providing mixed mode display.

To view mixed mode disassembly click [View mixed mode] button on this view.

Configuration of Disassembly window



S/W Breakpoints - ASM column — Display the PC location ( ), breakpoints ( ). Setting PC breakpoint by double-click.

Option

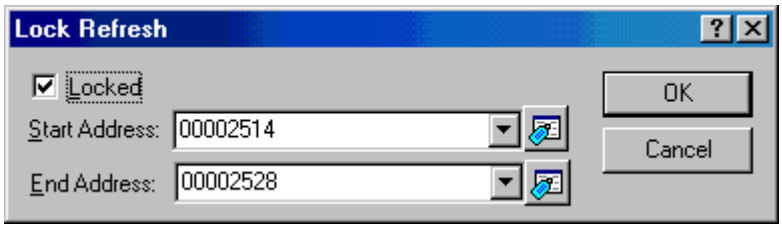
Clicking the right-hand mouse button displays a pop-up menu containing available options.

Pop-up menu options	Function
Lock Refresh...	It is possible to lock an memory range in the disassembly so that it does not refresh.
View Source	Launch editor at location in source.
Go To Cursor	Commences to execute the user program starting from the current PC address. The program will continue to run until the PC reaches the address indicated by the text cursor (not the mouse cursor) or another break condition is satisfied.
Set PC Here	Change the value of the PC to the address indicated by the text cursor (not the mouse cursor).
Set Address...	Enter a new start address.
Edit...	Modify the instruction at that address.
Find in Range...	Searches the range for the specified text string.
Copy	Places a copy of the highlighted text into the Windows® clipboard.
Label column width	Changes the column width of a label.
Save Disassembly Text...	Saves the specific range.
Print...	Prints the specific range.
Toggle Breakpoint	Sets or clears a software breakpoint at the line showing the address.
Enable/Disable Breakpoint	Enable or disable the current software breakpoint.

15.2.3 Disassembly lock refresh

It is possible to lock an memory range in the disassembly so that it does not refresh. This function is called the disassembly "lock refresh".

Right click on the disassembly window and select [Lock Refresh...]. The Lock Refresh dialog is displayed.



Check the [Locked] check box. The controls should now enable.

Select the start and end address that should be locked and cached so that no updates are displayed. Click OK. The view refreshes to only show the locked area.

15.2.4 Looking at a specific address

When you are looking at your program in a **Disassembly** view, you may want to look at another area of your program's code. Rather than scrolling through a lot of code in the program, you can go directly to a specific address. Select the [Set Address] option from the pop-up menu, and the [Set Address] dialog will be displayed.

Enter the address or label name in the edit box and either click the **OK** button or press the ENTER key. The **Disassembly** view updates to show the code at the new address. When an overloaded function or a class name is entered, the **Select Function** dialog box opens for you to select a function (Support for this function depends on the debugging platform.). For further information on this dialog, see the Selecting a function topic.

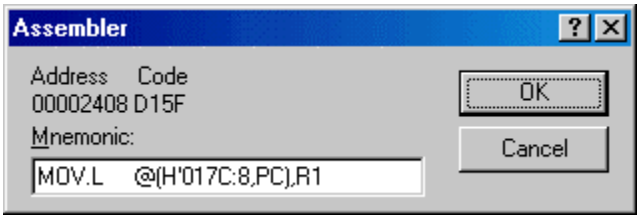
15.2.5 Looking at the current program counter address

Wherever you can enter an address or value into the HEW, you can also enter an expression. If you enter a register name prefixed by '#' (when the SH, H8, or R8C (E7/E8) family is in use) or '%' (when the M32C, M32R, M16C, or R8C (excluding E7/E8) family is in use), the contents of that register will be used as the value in the expression. Therefore if you open the **Set Address** dialog box and enter the expression '#PC' or '%PC', the **Source** or **Disassembly** view display will go to the current PC address. It also allows you to go to an address using an offset from the current PC by entering an expression including the PC register and an offset, e.g. '#PC+0x100' or '%PC+0x100'.

Sometimes it is difficult to know the exact location of the PC in your user program. To automatically display the PC click the Display PC toolbar button  , or select the [Debug->Display PC] menu item. This will open the editor or disassembly at the current PC.

15.2.6 Modifying assembly-language code

You can modify the assembly-language code in the disassembly view by double-clicking on the instruction that you wish to change. The **Assembler** dialog box will be displayed.



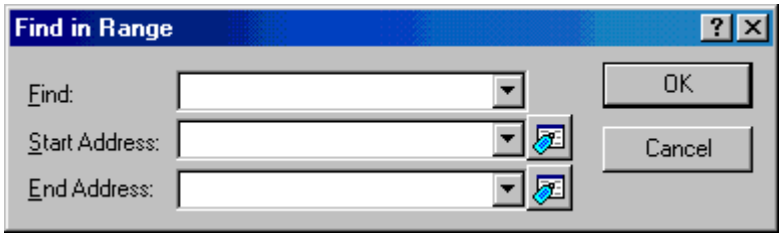
The address, machine code and disassembled instructions are shown. Type the new instruction or edit the old instruction in the **Mnemonic** field. Pressing ENTER will assemble the instruction into memory and move on to the next instruction. Clicking the **OK** button will assemble the instruction into memory and close the dialog box. Clicking the **Cancel** button or pressing ESC will close the dialog box.

Note:

The assembly-language display is disassembled from the actual machine code in the debugging platform's memory. If the memory contents are changed the dialog box (and **Disassembly** view) will show the new assembly-language code, but the source view will be unchanged. This is true even if the source file contains assembler.

15.2.7 Disassembly find in range

The disassembly find in range can be used to find a certain text string in the disassembly window between two addresses. Right click on the disassembly window and select [Find in range]. The [Find in range] dialog is displayed.



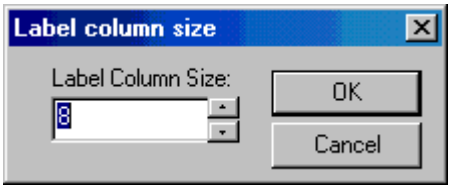
Enter your search string, the start and end address that should be searched. Click OK. The view then selects the first instance of that string in the range.

Note:

Subsequent find operations will find strings only in the paged disassembly area not the complete range.

15.2.8 Changing the Label Column Width

The column width of a label in the disassembly window can be changed by using the [Label column width] menu from the disassembly pop-up menu. The [Label column size] dialog is displayed.

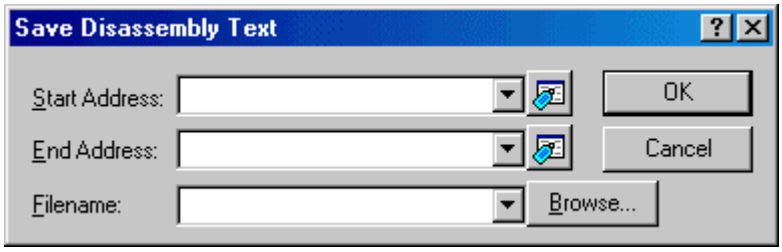


Specifies the column width (4-64) of a label. Click [OK].

15.2.9 Saving Disassembly Text

The contents of the disassembly window can be saved by using the [Save Disassembly Text...] menu from the disassembly pop-up menu.

When save is selected the [Save Disassembly Text] Dialog is displayed that asks you the range of addresses to save.

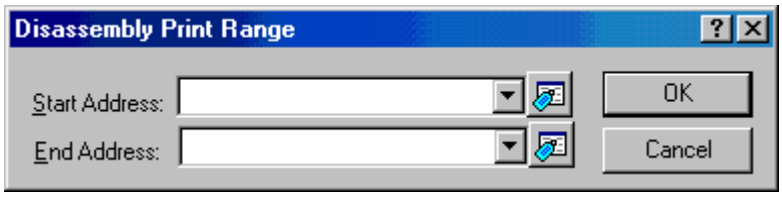


Both a start and end address should be supplied. You also need to specify the full filename to save the information to. If needed you can browse to the file to use.

15.2.10 Printing the disassembly view

The disassembly window can be printed by using the standard print menu or toolbar button when it is in focus or by using the menu [Print...] on the disassembly pop-up menu.

When print is selected the [Disassembly Print Range] Dialog is displayed that asks you the range of addresses to print.



Both a start and end address should be supplied.

Clicking OK on this dialog then passes the print selection to the standard print formatting and selection dialog. From here you can choose your printer and page setup options.

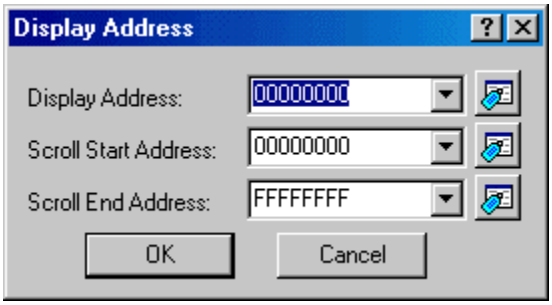
15.3 Operating memory

This section describes how to look at memory areas in the CPU's address space. How to look at a memory area in different formats, how to fill and move a memory block, and how to load and verify a memory area with a disk file are described.

15.3.1 Viewing a Memory Area

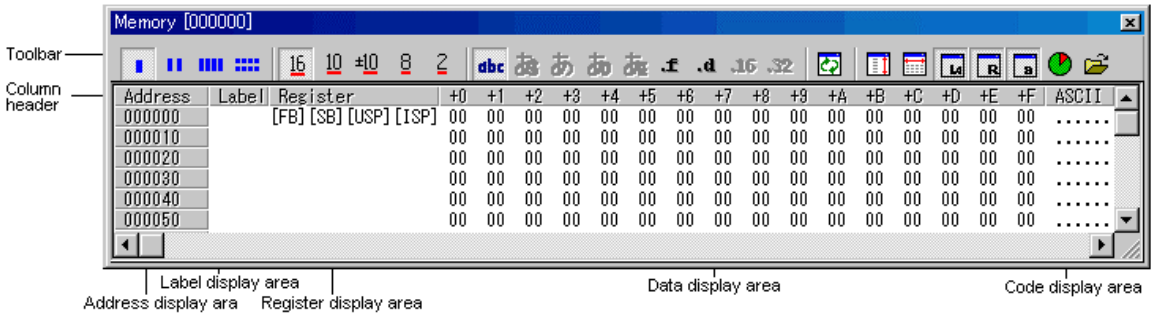
The Memory Window displays the contents of contiguous memory.

To open the **Memory** view, click the Memory toolbar button  , or select the [View–>CPU->Memory...] menu option.
You can specify the display start address and the scroll range at opening. The [Display Address] dialog box is opened. Specify the [Display Address], [Scroll Start Address] and [Scroll End Address].



Click the [OK] button or press the Enter key, and the dialog box closes and the [Memory] window opens. The display can be scrolled within the range of the entered display scroll start and end addresses.

Configuration of Memory window



- [+n] shows memory data read from [Address] and 'n' means the offset value from the first address of the row.
- [Code] shows the code names.
- [Register] shows the name of the register allocated to the first address of the memory data displayed on this row.
- In-place edit in the address display area/data display area/code display area.
- Double-clicking the address display area/label display area open a dialog, which allows you to change the display start address.
- A dialog which allows you to change the memory data at the clicked address by double-clicking the data display area/code data display area.

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of [Toolbar display] and [Customize toolbar...] are also included in the pop-up menu displayed by right-clicking the toolbar area.

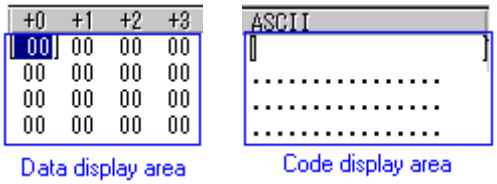
Pop-up menu options		Toolbar bottom	Function
Set...			Set data at specified address.
Fill...			Fill specified memory block with data.
Move...			Move specified memory block to.
Compare... *1			Comparing the contents of two memory blocks.
Test... *1			Testing an area of memory.
Save Memory contents...			Saving memory contents in a text file.
Search... *1			Finding a value in memory.
Search Next *1			Finding a next value in memory.
Address...			Specify display starting address.
Scroll Area...			Specify scroll range.
Register *1	(xxxxx)	-	Starting address to value of the register.
Followed Stack Pointer... *1			Keep tracking of the stack pointer position.
Set Start Up Symbol...			Changing the program display position Immediately after downloading.
Refresh			Update of the window contents.
Lock Refresh		-	Disabling update of the window contents.
Data Length	1byte		Display in 1-byte units.
	2bytes		Display in 2-bytes units.
	4bytes		Display in 4-bytes units.
	8bytes		Display in 8-bytes units.
Radix	Hex		Display in hexadecimal.
	Dec		Display in decimal.
	Signed Dec		Display in signed decimal.
	Oct		Display in octal.
	Bin		Display in binary.
Code	ASCII		Displaying memory as ASCII characters (default).
	SJIS/JIS/UNICODE/EUC	-	Not supported.
	Float		Displaying memory as single-precision floating point.
	Double		Displaying memory as double-precision floating point.
	16bit Fixed		Displaying memory as 16 bit fixed.
	32bit Fixed		Displaying memory as 32 bit fixed.
	24bit Accum	-	Displaying memory as 24 bit accumulate.
	40bit Accum	-	Displaying memory as 40 bit accumulate.
Layout	Label		Switch display or non-display of Label area.
	Register		Switch display or non-display of Register area.
	Code		Switch display or non-display of Code area.
Column...			Changing the number of digits displayed.
Coverage *1	Enable		Switching display or non-display of measurement result.
Save...			Saving memory contents in a file.
Load...			Loading a memory area from a file.
Split		-	Saving an area of memory.
Toolbar display		-	Showing/hiding toolbar buttons.
Customize toolbar...		-	Customizing toolbar buttons.

***1 :** Support for this function depends on the debugging platform.

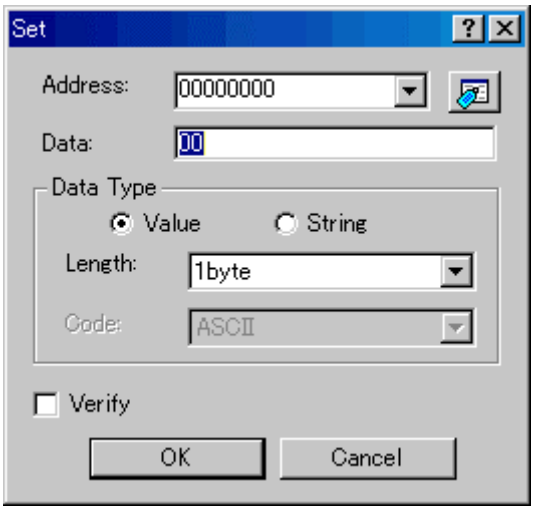
15.3.2 Modifying the Memory Contents

To change the contents of memory, follow the procedure below.

1. In-place edit in the data display area/code display area.



2. To change the contents of memory, open the **Set** dialog by selecting one of the following operations:
   - Double-click the data display area/code display area you want to change, **OR**
   - Select the data you want to change and choose the [Set...] option from the pop-up menu.



Enter the value (value or character) to be set in the [Data] field. Select the [Verify] check box. Support for verify function depends on the debugging platform.

When setting the value

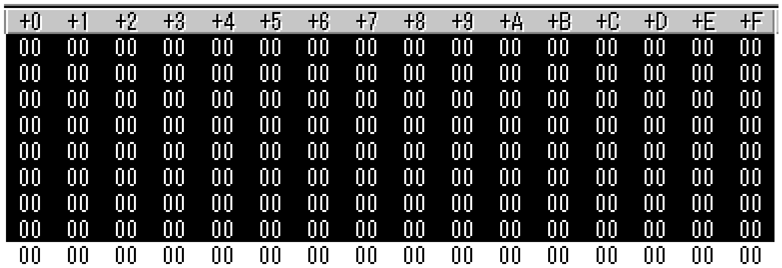
Click the [Value] button in the [Data Type] group. Specify the data length in the [Length] field.

When setting the character

Click the [String] button in the [Data Type] group. Specify the character code in the [Code] field.

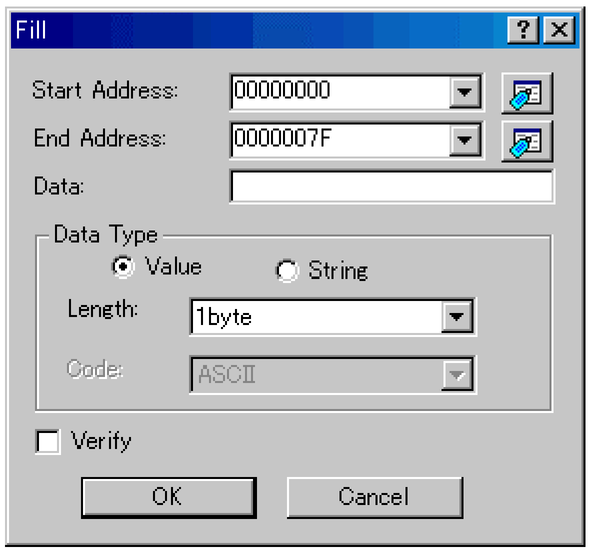
15.3.3 Selecting a Memory Range

If the memory address range is in the **Memory** view, you can select the range by clicking on the first memory unit (depending on your **Memory** view display choice) and dragging the mouse to the last unit. The selected range is highlighted.



15.3.4 Filling an Area of Memory with Constant Data

You can set the contents of a range of memory addresses to a value using the memory fill feature. Select an address range to fill in the [Memory] window by dragging the mouse. Choose the [Fill] option from the pop-up menu of the memory window. The [Fill] dialog box is displayed.



Enter the data (value or character) to be filled in the [Data] field.
Select the [Verify] check box. Support for verify function depends on the debugging platform.
If you did not drag the address range to be filled, you must enter the start/end address. The end address can also be prefixed by a plus (+); the end address will become the (start address) + (entered value).

When specifying the value

Click the [Value] button in the [Data Type] group. Specify the data length in the [Length] field.

When setting the character

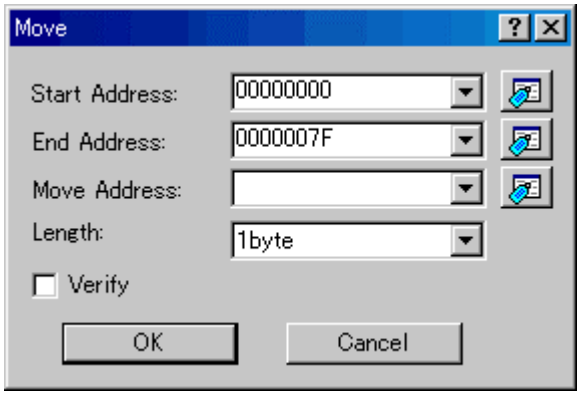
Click the String button in the [Data Type] group. Specify the character code in the [Code] field.

When the display data length is two bytes, two bytes' worth of a character can be specified.

Please use the [Set] dialog to specify the character string. (Select menu [Set...])

15.3.5 Copying an Area of Memory

You can copy an area of memory using the memory copy feature. Select a copy-source address range in the [Memory] window by dragging the mouse. Choose the [Move] option from the pop-up menu of the memory window. The [Move] dialog box is displayed.



Enter the copy destination start address in the [Move Address] field.

Select the [Verify] check box. Support for verify function depends on the debugging platform.

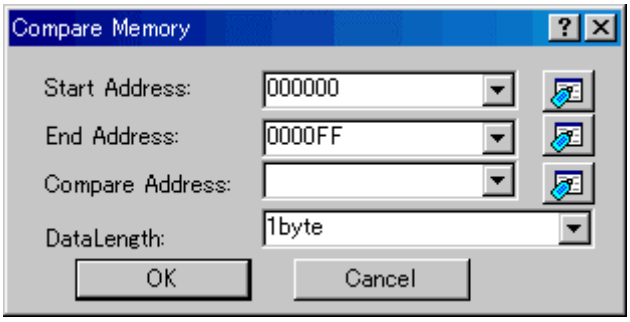
If you did not drag the copy-source address range, you must enter the start/end address. The end address can also be prefixed by a plus (+); the end address will become the (start address) + (entered value).

Drag & Drop

Type of dropped data	Operation
Selected range on the Memory Window's Data area	Copy the contents of a selected range of data to an area starting from the dropped position.

15.3.6 Comparing the Memory Contents

The contents of two memory blocks can be compared. Select a source address range in the [Memory] window by dragging the mouse. Choose the [Compare…] option from the pop-up menu of the memory window. The [Compare Memory] dialog box is displayed.



Enter the start address of the destination memory area in the [Compare Address] field and the data length in the [Data Length] field. If you did not drag the comparison-source address range, you must enter the start/end address. The end address can also be prefixed by a plus (+); the end address will become the (start address) + (entered value).

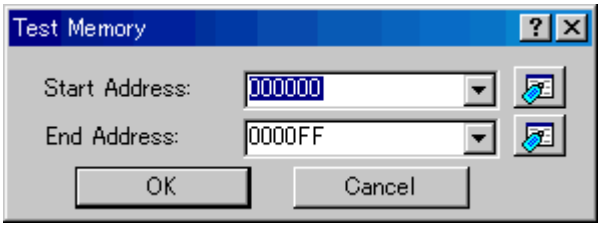
If there is a mismatch, the address where it was found is displayed in a message box.

Note:

Support for this function depends on the debugging platform.

15.3.7 Testing an Area of Memory

You can test an area of memory in the address space using the **Memory Test** feature. Select an address range to test in the [Memory] window by dragging the mouse. Choose the [Test] option from the pop-up menu of the memory window. The [Test Memory] dialog box is displayed.



If you did not drag the address range to be tested, you must enter the start/end address. The end address can also be prefixed by a plus (+); the end address will become the (start address) + (entered value).

Note:

The exact test is target dependent. However, in all cases the current contents of the memory will be overwritten – **YOUR PROGRAM OR DATA WILL BE ERASED**.

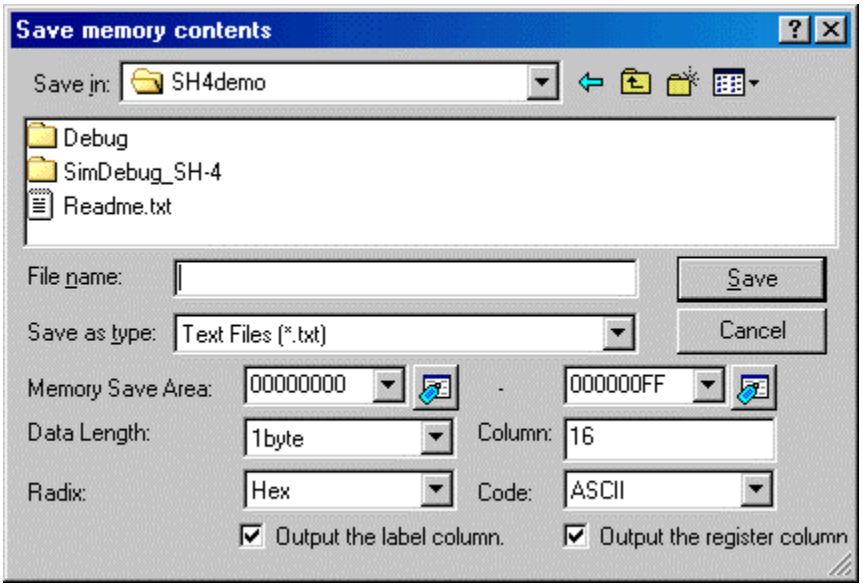
Support for this function depends on the debugging platform.

15.3.8 Saving Memory Contents in a Text File

You can save an area of memory in the address space to a text file using the **Save Memory Contents** feature. Select an address range to save in the [Memory] window by dragging the mouse. Choose the [Save Memory contents...] option from the pop-up menu of the memory window. The [Save memory contents] dialog box is displayed.

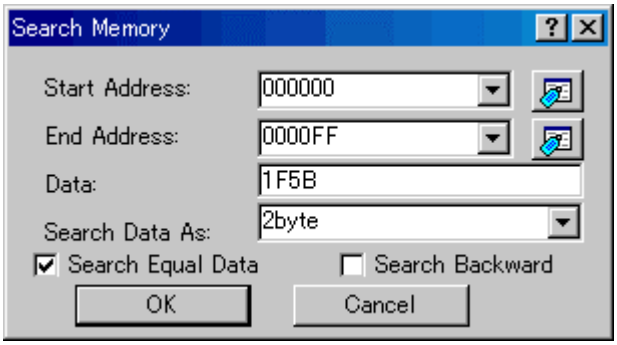
Select the output range in [Memory Save Area], data format in [Data Length], number of digits in [Column], radix in [Radix], and code in [Code]. It is possible to select showing/hiding of the label display area and register display area by [Output the label column] and [Output the register column], respectively.

If you did not drag the address range to be saved, you must enter the output range.



15.3.9 Finding a value in memory

You can find a value in memory using the **Search Memory** feature. Select an address range to search in the [Memory] window by dragging the mouse. Choose the [Search...] option from the pop-up menu of the memory window. The [Search memory] dialog box is displayed.



Enter a value you want to find in [Data] and select the data format in [Search Data As]. Check [Search Equal Data] to find the exact value specified. To change the search direction, check [Search Backward].

If [Pattern search] is selected as the search format, a byte string of up to 256 bytes can be searched for.

The end address can also be prefixed by a plus sign (+), which will use the entered value as a range.

If the data could not be found, the [Memory] window display remains unchanged and a message box informing that the data could not be found is displayed.

If [Search Next] is selected from the pop-up menu in the state where data has been found, the search will continue from the next address.

Note:

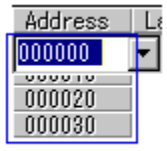
Support for this function depends on the debugging platform.

15.3.10 Changing the Display Address

Use the scroll bar, Page Up/Page Down key and Up/Down key to change the display position.

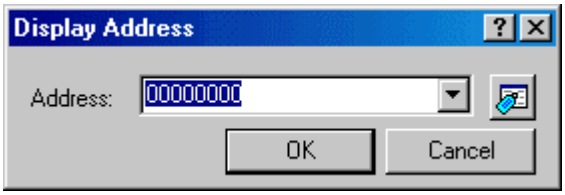
To change the display address directly, follow the procedure below.

1. In-place edit in the address display area.



Address display area

2. To change the display address, open the **Display Address** dialog by selecting one of the following operations:
   - Double-click the address display area you want to change, **OR**
   - Choose the [Address...] option from the pop-up menu.



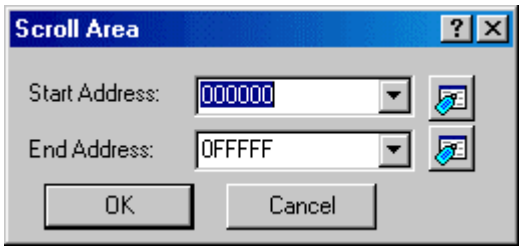
Specify the displaying address in the [Address] field.

Drag & Drop

Manipulation	Operation
Select address on Memory Window's Address area and drop it into another Memory Window's Address area	Changes the window's display start address to that address.
Select address (text string) on Disassembly Window's Address area and drop it into Memory Window's Address area	Changes the window's display start address to that address.
Select variable name (text string) and drop it into Memory Window's Address area	Changes the window's display start address to that address.

15.3.11 Changing the Scroll Area

Select the [Scroll Area...] menu from the pop-up menu of the memory window. The [Scroll Area] dialog is displayed.



Specify the scroll range to be displayed. By default, the scroll range is set to 0 to the maximum address of MCU.

15.3.12 Starting address to value of the register

Select the register from the followings in the pop-up menu [Register] of the memory window.

Note:

Support for this function depends on the debugging platform.

15.3.13 Tracking the stack pointer position

The memory window has a function that alters the display address while tracking the stack pointer position (By default, the display does not track the stack pointer position.). To track the stack pointer position, choose [Followed Stack Pointer...] option from the pop-up menu of the memory window. The [Followed Stack Pointer] dialog box is displayed.

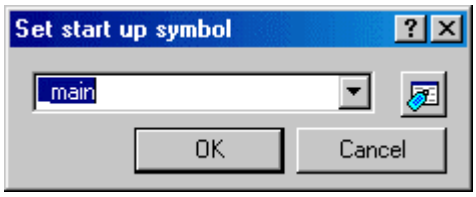
Select the stack pointer to track from the drop-down list box. The Memory Window has its display addresses automatically altered by tracking the selected stack pointer position. Selecting "Not Following" in the [Followed Stack Pointer] dialog box causes the debugger to stop tracking the stack pointer position.

Note:

Support for this function depends on the debugging platform.

15.3.14 Changing the Program Display Position Immediately After Downloading

To specify the source file position , select [Set Start Up Symbol...] option from pop-up menu of the memory window. The [Set start up symbol] dialog box is displayed.



Input start up symbol to drop-down list box.

15.3.15 Updating the Window Contents

The [Memory] window contents can be forcibly updated. Selecting the [Refresh] from the pop-up menu of the memory window.

15.3.16 Disabling Update of the Window Contents

Automatic update of the [Memory] window contents, which is performed when user program execution stops and in other cases, can be disabled. The [Memory] window will be grayed.

Selecting the [Lock Refresh] from the pop-up menu of the memory window.

15.3.17 Changing the Data Length

Select the data length from the followings in the pop-up menu [Data Length] of the memory window.

Either the following can be specified.

[1byte]	Display in 1-byte units (default).
[2bytes]	Display in 2-bytes units.
[4bytes]	Display in 4-bytes units.
[8bytes]	Display in 8-bytes units.

15.3.18 Changing the Radix

Select the data radix from the followings in the pop-up menu [Radix] of the memory window.

Either the following can be specified.

[Hex]	Display in hexadecimal (default).
[Dec]	Display in decimal.
[Signed Dec]	Display in signed decimal.
[Oct]	Display in octal.
[Bin]	Display in binary.

15.3.19 Changing the Code

Select the code from the followings in the pop-up menu [Code] of the memory window.

Either the following can be specified.

[ASCII]	Displaying memory as ASCII characters (default).
[SJIS/JIS/UNICODE/EUC]	Not support.
[Float]	Displaying memory as single-precision floating point.
[Double]	Displaying memory as double-precision floating point.
[16bit Fixed]	Displaying memory as 16bit fixed.
[32bit Fixed]	Displaying memory as 32bit fixed.
[24bit Accum]	Displaying memory as 24bit accumulate.
[40bit Accum]	Displaying memory as 40bit accumulate.

15.3.20 Setting the Layout

Select the layout from the followings in the pop-up menu [Layout] of the memory window.

The followings can be selected:

[Label]	Switch display or non-display of label.
[Register]	Switch display or non-display of Register.
[Code]	Switch display or non-display of Code.

When the label, register or code is shown, the option is checked.

When the label, register and code are hidden:

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

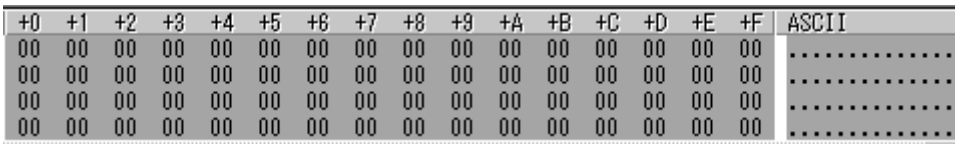
15.3.21 Changing the number of digits displayed

Choose [Column...] menu from the pop-up menu of the memory window. The [Set Column] dialog box is displayed.

Specify the number of digits in which you want data to be displayed (1-256).

15.3.22 Switching display or non-display of Measurement result

In the memory window, a display of coverage measurement is set to "Disable" by default. To enable the display, select [Coverage->Enable] menu from the pop-up menu of the memory window. In the Memory window, the background of the executed lines is displayed in sky blue, and the background of the unexecuted lines is displayed in gray.



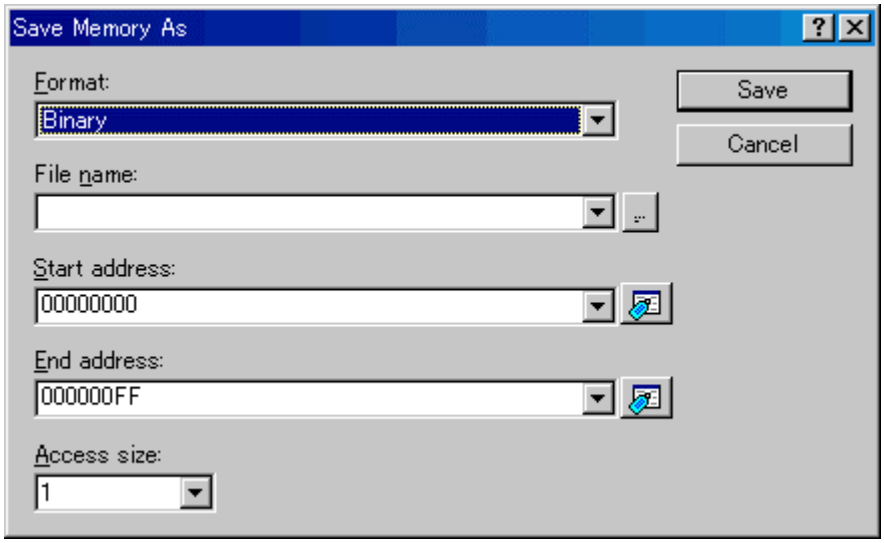
Note:

Support for this function depends on the debugging platform.

15.3.23 Saving an area of memory

You can save an area of memory in the address space to a disk file using the **Save Memory** feature. Select an address range to save in the [Memory] window by dragging the mouse. Choose the [Save...] option from the pop-up menu of the memory window. The [Save Memory As] dialog box is displayed

(This operation can also be achieved by selecting [Debug->Save Memory…].)



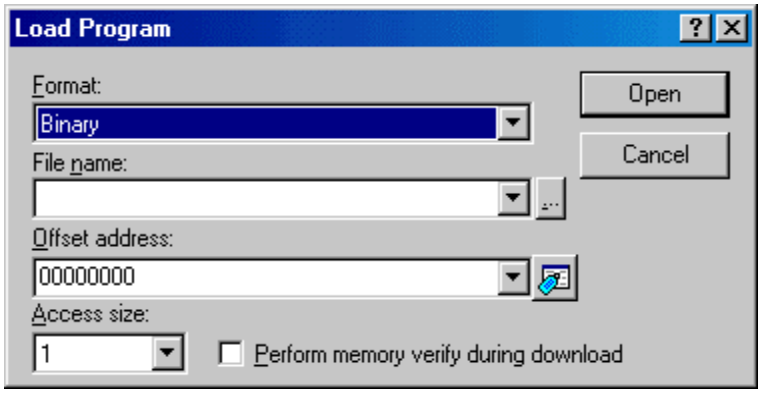
If you did not drag the address range to be saved, you must enter the start/end address. The end address can also be prefixed by a plus sign (+), which will use the entered value as a range.

Enter the file format in [Format], file name in [File name], and access size in [Access size]. The [File name] drop-down list contains the last four filenames used for saving memory. Alternatively, click the **Browse** button to launch a standard [Save As] dialog. The access size for saving data can be selected from the [Access size] drop-down list. When the data is saved in memory with little endian, the order of data depends on the access size.

When the file save is complete a confirmation message box will be displayed.

15.3.24 Loading a Memory Area from a File

A file can be loaded to the debugging platform's memory. Choose [Load…] from the pop-up menu of the memory window. The [Load Program] dialog box is displayed.



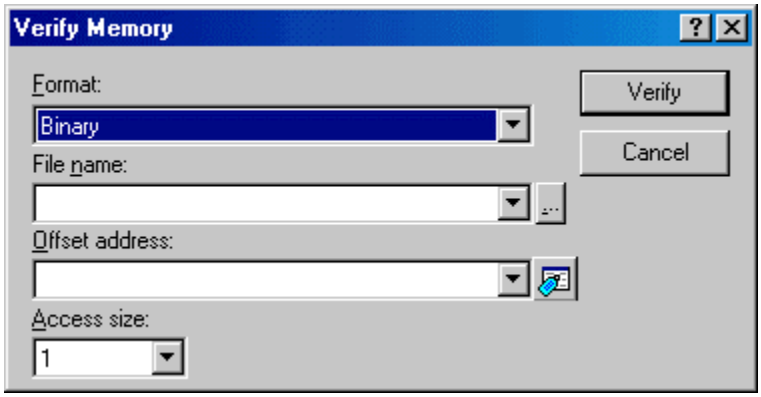
Enter the file format in [Format], file name in [File name], offset address in [Offset address], and access size in [Access size]. To verify memory, check [Perform memory verify during download].

15.3.25 Splitting Up the Window Display

To vertically divide the [Memory] window display into two, select [Split] from the pop-up menu and move the split-up bar.

15.3.26 Verifying a Memory Area

A memory area in the address space can be verified using the memory verify function. Choose the [Debug->Verify Memory...] option. The [Verify Memory] dialog box is displayed



Enter the file format in [Format], file name in [File name], offset address in [Offset address], and access size in [Access size].

Note:

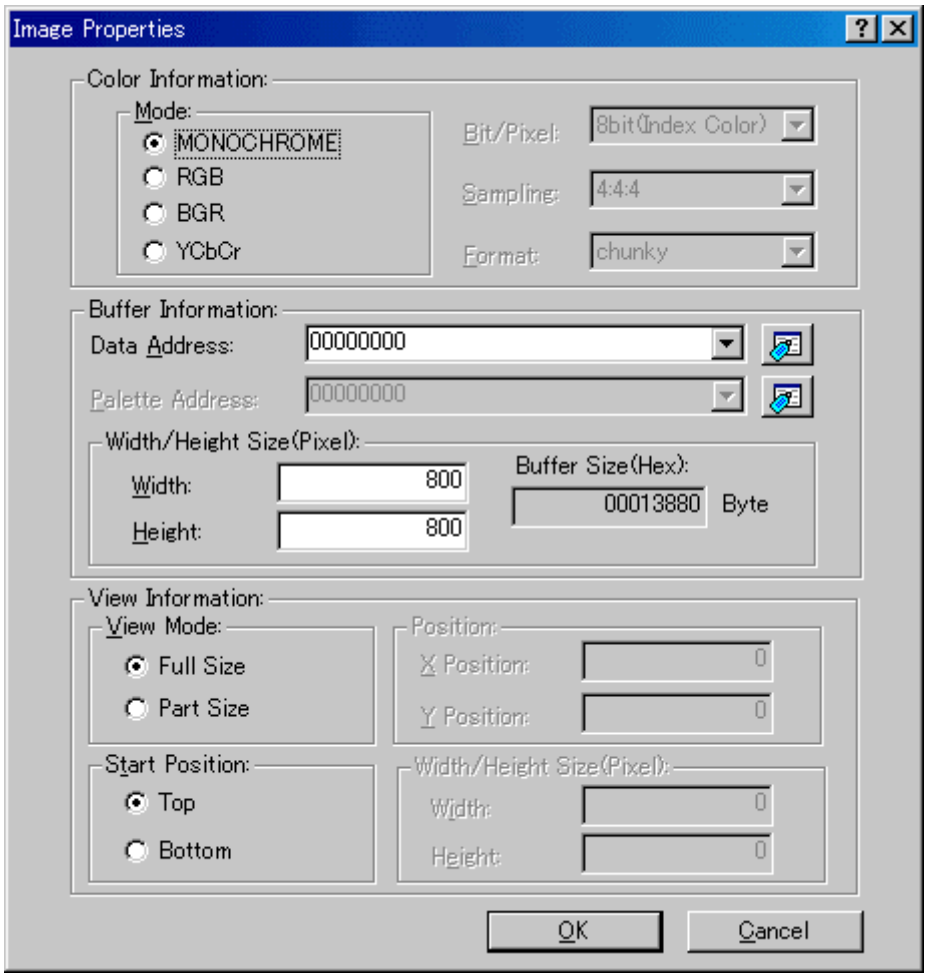
Support for this function depends on the debugging platform.

15.4 Displaying memory contents as an image

The memory contents can be displayed as an image in the [Image View] window.

15.4.1 Opening the Image View Window

Click the [Image] toolbar button [] or choose [View->Graphic->Image…] to open the [Image Properties] dialog box.



The [Image Properties] dialog box is used to specify the display method of the [Image View] window.

The following items are to be specified:

- [Color Information] Specifies the color information of the image to be displayed.
  - [Mode] Specifies the format.
    - [MONOCHROME] Displays in black and white.
    - [RGB] Displayed in R (red), G (green), and B (blue)
    - [BGR] Displayed in B (blue), G (green), and R (red)
    - [YCbCr] Displayed by Y (brightness), Cb (color difference in blue), and Cr (color difference in red)
  - [Bit/Pixel] Specifies Bit/Pixel according to the selected [Mode]. (Valid when RGB or BGR is selected)
  - [Sampling] Specifies the format of sampling. (Valid when YCbCr is selected)
  - [Format] Specifies Chunky/planar. (Valid when YCbCr is selected)
- [Buffer Information] Specifies the area to store data, size, and the address of the palette.
  - [Data Address] Specifies the start address of the memory where image data is to be displayed. (Displayed in hexadecimal)
  - [Palette Address] Specifies the start address of the memory of color palette data. (Displayed in hexadecimal) (Valid when 8Bit is selected for RGB or BGR)
  - [Width/Height Size] Specifies the width and height of the image.
    - [Width (Pixel)] Specifies the width of the image. (When a prefix is omitted, the values are input and displayed in decimal.)
    - [Height (Pixel)] Specifies the height of the image. (When a prefix is omitted, the values are input and displayed in decimal.)
    - [Buffer Size] Displays the buffer size of the image from the width and height (Displayed in hexadecimal)
- [View Information] Specifies the location, size, and data start location of the part to be displayed among the entire image.
  - [View Mode] Specifies the entire/part to be displayed in the image.
    - [Full Size] Displays the entire image.
    - [Part Size] Displays part of the image.
  - [Start Position]
    - [Top] Displays data from the upper left.
    - [Bottom] Displays data from the lower left.
  - [Position] Specifies the start position of the image where part of the image is to be displayed. (Valid when [Part Size] is selected)
    - [X Position] Specifies the X axis of the start location. (When a prefix is omitted, the values are input and displayed in decimal.)
    - [Y Position] Specifies the Y axis of the start location. (When a prefix is omitted, the values are input and displayed in decimal.)
  - [Width/Height Size] Specifies the height and width of the image to be displayed partly.
    - [Width (Pixel)] Displays the width of display. (When a prefix is omitted, the values are input and displayed in decimal.)

[Height (Pixel)]	Displays the height of display. (When a prefix is omitted, the values are input and displayed in decimal.)

After the settings have been made in the [Image Properties] dialog box, clicking the [OK] button opens the [Image View] window.

Even after the [Image View] window is displayed, the display contents can be modified by opening this dialog box by choosing [Properties…] from the pop-up menu.

Displays the memory contents as an image.

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

Pop-up menu options		Function
Auto Refresh	Nonrefresh	Not refresh the window contents.
	Stop	Automatically update the window contents when user program execution stops.
	Real time	Real-time updated the window contents.
Refresh Now		Updates the window contents.
Properties...		Displays information on the pixel on which the mouse pointer is located.

15.4.2 Automatically Updating the Window Contents

Checking [Auto Refresh->Not Refresh] in the pop-up menu will not refresh the window.

Checking [Auto Refresh->Stop] in the pop-up menu will allow the window contents to be automatically updated when user program execution stops.

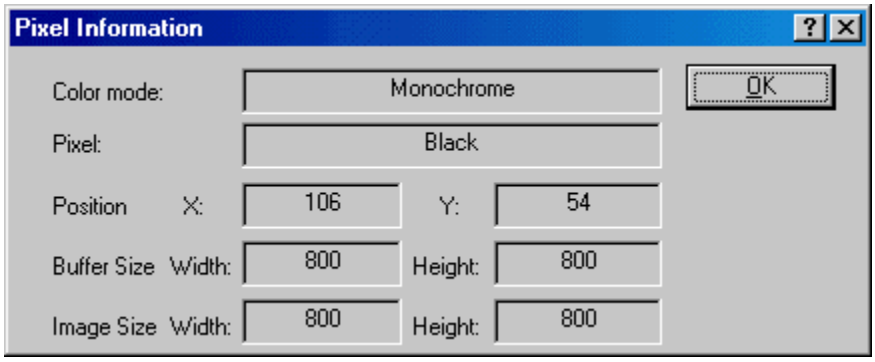
Checking [Auto Refresh->Realtime] in the pop-up menu will allow the window contents to be real-time updated.

15.4.3 Updating the Window Contents

Selecting [Refresh Now] from the pop-up menu immediately updates the window contents.

15.4.4 Displaying the Pixel Information

Double-clicking within the window displays information on the pixel on which the mouse pointer is located in the [Pixel Information] dialog box.



This dialog box displays pixel information on the cursor location.

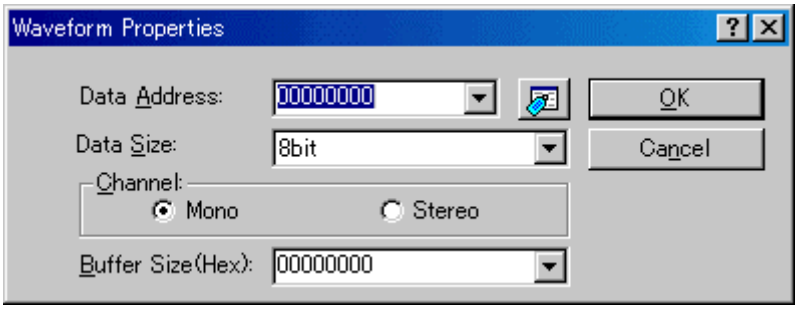
[Color Mode]	Displays the format of the image.	
[Pixel]	Displays color information of the cursor location. (Displayed in decimal)	
[Position]	Displays the cursor location in X and Y axis. (Displayed in decimal)	
	[X]	Displays the X axis of the cursor location.
	[Y]	Displays the Y axis of the cursor location.
[Buffer Size]	Displays the buffer size. (Displayed in decimal)	
	[Width]	Displays the buffer width.
	[Height]	Displays the buffer height.
[Image Size]	Displays the width and height of the display. (Displayed in decimal)	
	[Width]	Displays the width.
	[Height]	Displays the height.

15.5 Displaying memory contents as waveforms

Memory contents can be displayed as wave forms in the [Waveform View] window.

15.5.1 Opening the Waveform View Window

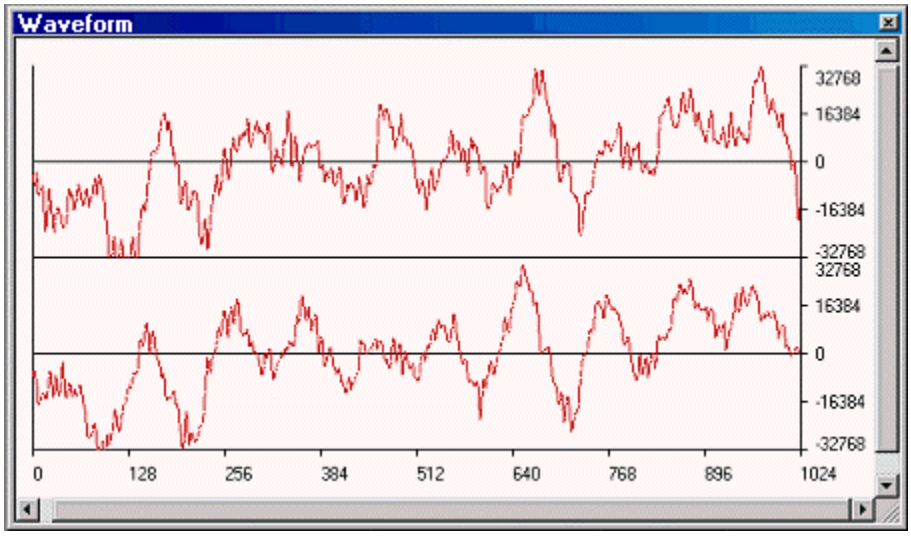
Choose [View->Graphic->Waveform…] or click the [Wave form] toolbar button [ ] to open the [Waveform Properties] dialog box.



Specifies the waveform format. The following items can be specified.

[Data Address]	Specifies the start address of data in memory. (Displayed in hexadecimal)
[Data Size]	Selects 8Bit or 16Bit.
[Channel]	Specifies Mono or Stereo.
[Buffer Size]	Specifies the buffer size of data. (Displayed in hexadecimal)

After the settings have been made in the [Waveform Properties] dialog box, clicking the [OK] button opens the [Waveform View] window.



Even after the [Waveform View] window is displayed, the display contents can be modified by opening this dialog box by choosing [Properties…] from the pop-up menu.

Displays the memory contents as waveforms. The X axis shows the number of sampling data and the Y axis shows the sampling value.

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

Pop-up menu options		Function
Auto Refresh	Nonrefresh	Not refresh the window contents.
	Stop	Automatically update the window contents when user program execution stops.
	Real time	Real-time updated the window contents.
Refresh Now		Updates the window contents.
Zoom In		Zoom-in display.
Zoom Out		Zoom-out display.
Reset Zoom		Resets the zoom display.
Zoom Magnification	X2	The zoom magnification is 2.
	X4	The zoom magnification is 4.
	X8	The zoom magnification is 8.
Scale	128	The size of the X axis is 128 pixels.
	256	The size of the X axis is 256 pixels.
	512	The size of the X axis is 512 pixels.
Clear Cursor		Hides the cursor display.
Sample Information...		Displays the sampling information of the cursor location.
Properties...		Specifies the waveform format.

15.5.2 Automatically Updating the Window Contents

Checking [Auto Refresh->Not Refresh] in the pop-up menu will not refresh the window.

Checking [Auto Refresh->Stop] in the pop-up menu will allow the window contents to be automatically updated when user program execution stops.

Checking [Auto Refresh->Realtime] in the pop-up menu will allow the window contents to be real-time updated.

15.5.3 Updating the Window Contents

Selecting [Refresh Now] from the pop-up menu immediately updates the window contents.

15.5.4 Zoom-In Display

Selecting [Zoom In] from the pop-up menu displays the waveforms with the horizontal axis enlarged.

15.5.5 Zoom-Out Display

Selecting [Zoom Out] from the pop-up menu displays the waveforms with the horizontal axis reduced.

15.5.6 Resetting the Zoom Display

Selecting [Reset Zoom] from the pop-up menu displays the waveforms in its original size.

15.5.7 Setting the Zoom Magnification

In the [Zoom Magnification] submenu of the pop-up menu, the zoom magnification can be selected from 2, 4, or 8.

15.5.8 Setting the Horizontal Scale

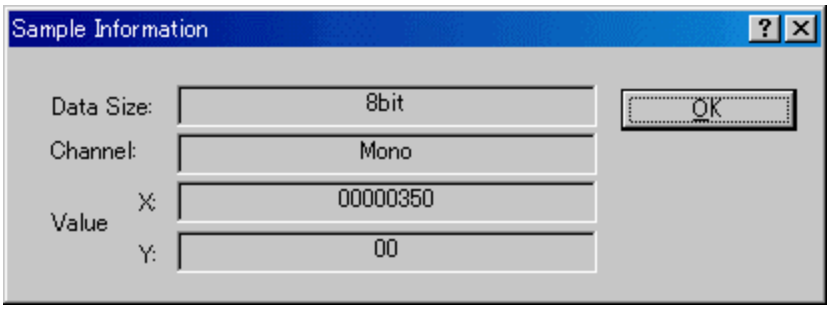
In the [Scale] submenu of the pop-up menu, the size of the X axis can be selected from 128, 256, or 512 pixels.

15.5.9 Non-Display of Cursor

Selecting [Clear Cursor] from the pop-up menu hides the cursor display.

15.5.10 Displaying the Sampling Information

Selecting [Sample Information…] from the pop-up menu displays the [Sample Information] dialog box.



Displays the sampling information of the cursor location in the [Waveform View] window. The following information is displayed.

[Data Size]	Displays 8bit or 16bit.	
[Channel]	Displays the data channel.	
[Value]	[X]	Displays the X axis of cursor location.
	[Y]	Displays the Y axis of cursor location (displays Y axes for both the upper and lower plots when Stereo is selected).

15.6 Viewing the IO memory

As well as a CPU and ROM/RAM, a micro-controller also contains on-chip peripheral modules. The exact number and type of peripheral modules differ between devices but typical modules are DMA controllers, serial communications interfaces, A/D converters, integrated timer units, a bus state controller and a watchdog timer. Accessing registers, which are mapped to the micro-controller's address space, programs the on-chip peripherals.

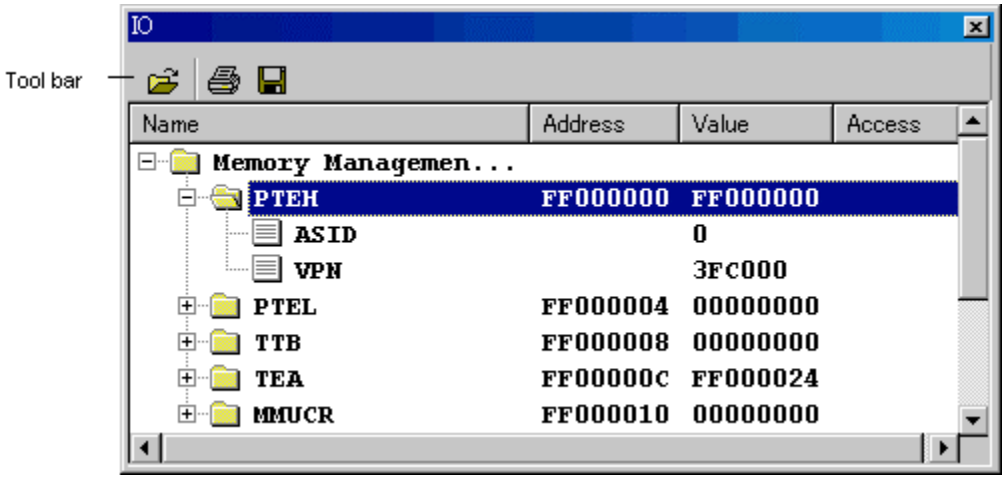
Since the setting up and use of these on-chip peripheral registers is usually very important in an embedded micro-controller application, it is useful to be able to look clearly at the contents of these registers. The **Memory** view only allows you to look at data in memory as byte, word, long word, single-precision floating-point, double-precision floating-point, or ASCII values, so the HEW also provides an **I/O Registers** view to ease the inspection and setting up of these registers.

To open an **I/O Registers** view, select the [View–>CPU->IO] menu option, or click the View I/O toolbar button  . Modules that match the on-chip peripherals organize the I/O register information. When an **I/O Registers** view is first opened, only a list of module names is displayed.

15.6.1 Opening the IO Window

To open the [IO] window, select [View->CPU->IO] or click the [View IO] toolbar button []. Modules that match the on-chip peripheral modules organize the I/O register information. When the [IO] window is first opened, only a list of module names is displayed.

Configuration of IO window



- Expanding the I/O register/bit display.
- If the I/O register/bit value is changed, the value is displayed in red.
- Double-clicking the I/O register/bit display line opens a dialog, which allows you to change a register value.
- The I/O register/bit contents line can be changed by in-place editing.

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of [Toolbar display] and [Customize toolbar...] are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar bottom	Function
Load IO File...		Manually load an IO file.
Print		Prints the contents currently displayed in the window.
Save To File...		Saves the contents currently displayed in the window in a text file.
Toolbar display	-	Showing/hiding toolbar buttons.
Customize toolbar...	-	Customizing toolbar buttons.

15.6.2 Expanding an I/O register display

To display the names, addresses and values of the I/O registers, double-click on the module name or select the module name, by clicking on it or using the cursor keys, and press the cursor right key. The module display will expand to show the individual registers of that peripheral module and their names, addresses and values. Double-clicking (or pressing the cursor left key) again on the module name will close the I/O register display.

Note:

If you are using an emulator-based debugging platform, reading data from an I/O register can sometimes affect the operation of your program. For example, reading a data register can cancel a pending interrupt. Data is only read from I/O modules that have been expanded in the [IO] window (so that the register values are displayed). Therefore, as long as I/O modules are collapsed when they no longer need to be displayed, this will not cause a problem. Also, note that having a [Memory] window (or [Disassembly] window) open on the I/O area can have the same effect.

15.6.3 Manually loading an IO file

To manually load an IO file right click on the IO window and select the [Load IO File…] menu item on the IO window pop-up. A standard file browser is invoked. Simply select the file you require to load and click OK. The IO file will be loaded into the window.

See Reference 5, I/O File Format, for more information about a IO file format.

15.6.4 Printing the Currently Displayed Contents

The contents currently displayed in the window can be printed in a text file. Select [Print] from the pop-up menu.

15.6.5 Saving the Currently Displayed Contents

The contents currently displayed in the window can be saved in a text file. Select [Save to File…] from the pop-up menu.

15.6.6 Modifying I/O register contents

To edit the value in an I/O register you can type hex values directly into the view. To enter more complex expressions, double-click or press ENTER on the register to open a dialog box to modify the register contents. When you have entered the new number or expression, click the **OK** button or press ENTER. The dialog box closes and the new value is written into the register.

15.7 Looking at registers

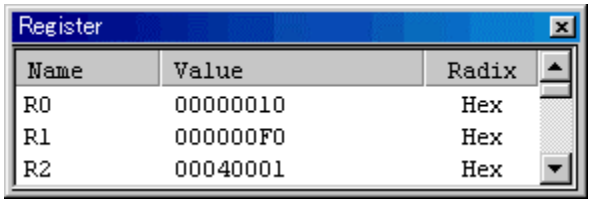
If you are debugging at assembly-language level using the **Disassembly** view, you will probably find it useful to see the contents of the CPU's general registers. You can do this by using the **Registers** view.

15.7.1 Opening the Register Window

The Register window displays the register data and flag data. You can change a register/flag value from the window.

To open the [Register] window, click the Registers toolbar button  , or select the [View–>CPU->Register] menu option. The [Register] window opens showing all of the CPU's general registers and values (displayed in hexadecimal).

Configuration of Register window



- If a register/flag value is changed, the value is displayed in red.
- Double-clicking the register display line opens a dialog, which allows you to change a register value.
- You can change a flag value by clicking the button corresponding to the flag.
- The right-click menu allows you to change the display radix point and the register bank (Change of the register bank can be selected only when the target platform supports this function.).

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

Pop-up menu options		Function
Radix	Hex	Display in hexadecimal.
	Dec	Display in decimal.
	Oct	Display in octal.
	Bin	Display in binary.
Bank0 ***1**		Display registers of bank 0.
Bank1 ***1**		Display registers of bank 1.
Layout	Radix	Switches display or non-display of radix.
	FLAGs	Switches display or non-display of flags display area.
	Settings...	Chooses a register to be displayed.
Edit...		Changes a register's contents.
Split		Splits up the window Display.
Save To File...		Saves register contents in a text file.

***1 :** Support for this function depends on the debugging platform.

15.7.2 Changing the Register Display Radix

You can change the display radix by register.

To do this, click the mouse right button on the register to be changed and select the display radix from the pop-up menu which is opened.

The followings can be selected:

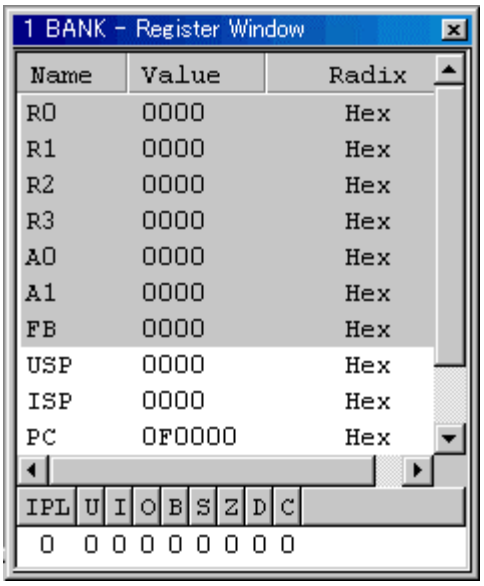
[Hex]: Display in hexadecimal.
[Dec]: Display in decimal.
[Oct]: Display in octal.
[Bin]: Display in binary.

15.7.3 Switching Register Bank

Immediately after opening the Register window, the register data for the bank corresponding to the value of flag is displayed.

To reference the register data of Bank1, select [Bank1] menu option with the Register window active.

The register specific to Bank1 is displayed in the gray background. (Example of M16C family debugger)



To reference the register data of Bank0, select [Bank0] menu option with the Register window active. (Through the operation of option [Bank0] and [Bank1], the value of flag does not change.)

To switch the bank, you can also use the pop up menu which is displayed by clicking the mouse right button on the register display area in the Register window, or change the value of flag.

(If you change the value of flag, the register bank also changes in response to the value.)

Note:

Support for this function depends on the debugging platform.

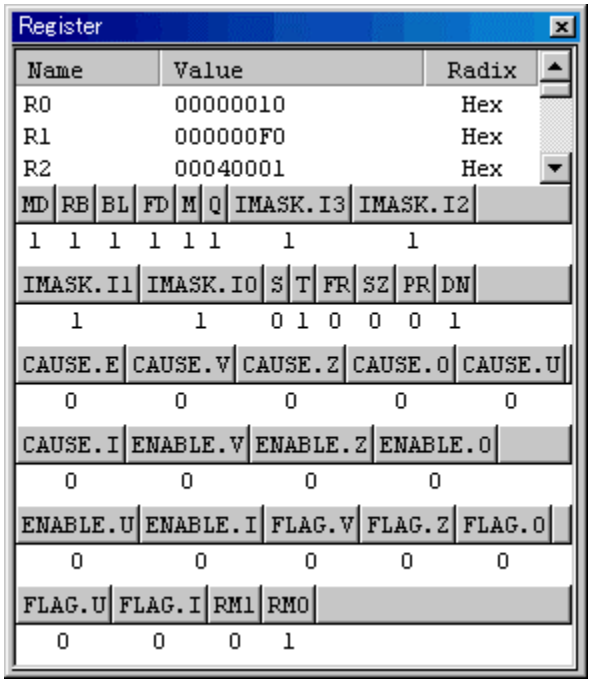
15.7.4 Setting the Layout

To set the layout of the Register Window, choose [Layout] menu from the register Window pop-up menu. The followings can be selected:

[Radix] Switch display or non-display of radix.
[FLAGs] Switch display or non-display of flags display area.

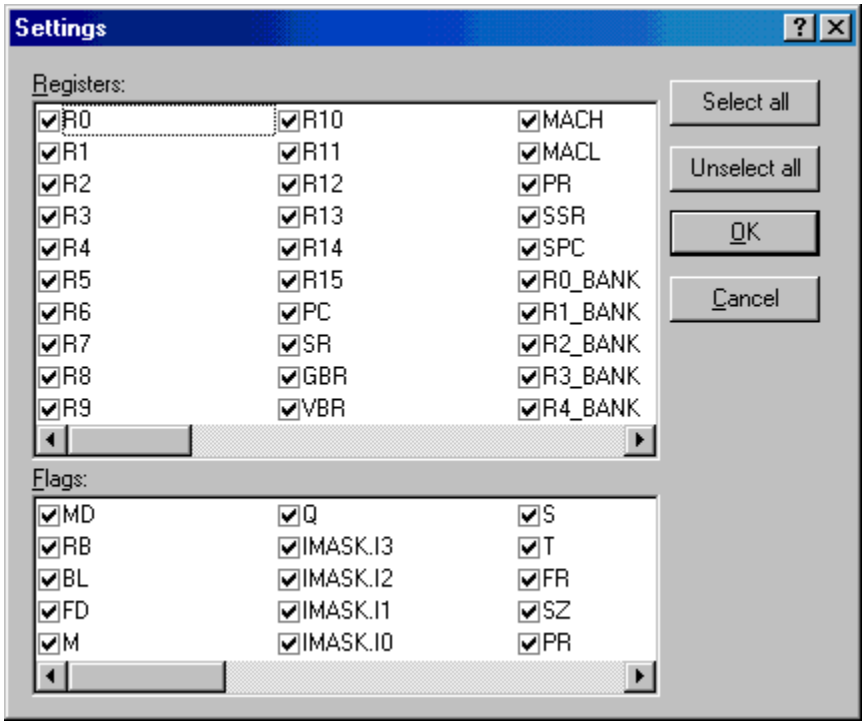
When the radix or flag is shown, the option is checked.

When the radix and flags are displayed:



15.7.5 Choosing a Register to be Displayed

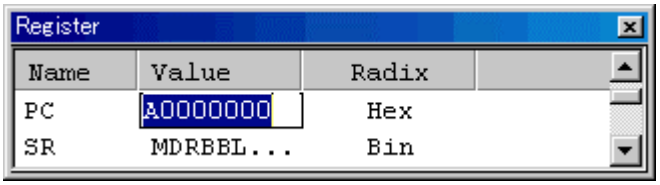
To choose a register to be displayed in the [Register] window, choose [Settings…] menu from the register pop-up menu. This dialog box is shown in following figure.



15.7.6 Modifying Register Contents

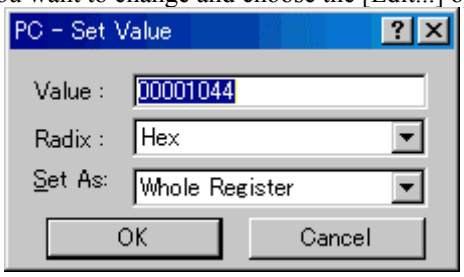
To change register contents, follow the procedure below.

Enter a value in the [Value] field of the register you want to change.



To change a register's contents, open the **Set Value** dialog by selecting one of the following operations:

- Double-click the register you want to change, **OR**
- Select the register you want to change and choose the [Edit...] option from the pop-up menu.



You can enter a number or C/C++ expression in the [Value] field.

You can choose the radix from the [Radix] drop-down list box.

You can choose whether to modify the entire contents of the register, a masked area, floating bits or flag bits, by selecting an option from the [Set As] drop-down list box (the contents of this list depends on the CPU model and selected register).

When you have entered the new number or expression, click the **OK** button or press ENTER. The dialog box closes and the new value is written into the register.

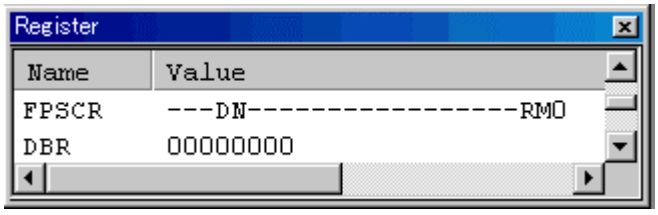
15.7.7 Setting the Flag Value

When the flag itself is displayed

Click the button of the flag to be changed. Every time you click the button the flag status (1/0) is switched. If a flag is composed of multiple bits, a dialog is opened, where you can enter a value to be changed.

When the flag is displayed in the register

Double-click the FLG line. A dialog is opened. Enter the value to be changed.



15.7.8 Splitting Up the Window Display

To vertically divide the [Register] window display into two, select [Split] from the pop-up menu and move the split-up bar. Moving the split-up bar to the top end or bottom end of the window cancels the split-up display.

15.7.9 Saving Register Contents

To save register contents in a text file, choose [Save To File...] menu from the register pop-up menu. The [Save As] dialog box shown below will be displayed. Specify the file name.

15.7.10 Using register contents

It can be useful to be able to use the value contained in a CPU register when you are entering a value elsewhere in the HEW, for example when displaying a specified address in the **Disassembly** or **Memory** views. You can do this by specifying the register name prefixed by the '#' (when the SH, H8, or R8C (E7/E8) family is in use) or '%' (when the M32C, M32R, M16C, or R8C (excluding E7/E8) family is in use) character, e.g.: '#PC' or '%PC'.

Note:

Support for this function depends on the debugging platform.

15.8 Resetting the Target MCU

To reset the target MCU, click the Reset CPU toolbar button  , or select the [Debug->Reset CPU] menu option.

Resetting the target MCU initializes the on-chip I/O registers and makes the program counter jump to the address set in the reset vector.

15.9 Set PC to Cursor

To changes the value of the PC to the address at the row of the text cursor, click the Set PC to the cursor button  , or select the [Debug->Set PC to the Cursor] menu option.

15.10 Initialize the Debugging Platform

Select the [Debug->Initialize] menu option.

It will close down any open child windows and shut down the link to the debugging platform. If this is successful, an attempt to re-establish the link to the debugging platform will be made.

15.11 Connects/Disconnects the Debugging Platform

To connects the debugging platform:

1. Click the Connect toolbar button  , or select the [Debug->Connect] menu option.

To disconnects the debugging platform:

1. Click the Disconnect toolbar button  , or select the [Debug->Disconnect] menu option.

Note:
Support for this function depends on the debugging platform.

15.12 Executing your program

This section describes how you can execute your program's code. You will learn how to do this by either running your program continuously or stepping single or multiple instructions at a time.

15.12.1 Continuing run

When your program is stopped and the debugger is in **Break Mode**, the HEW will display a yellow arrow in the gutter of the line in the **Editor** and **Disassembly** views that correspond to the CPU's current Program Counter (PC) address value. This will be the next instruction to be executed if you perform a step or continue running.

To continue running from the current PC address:

- Click the Continue toolbar button  , **OR**
- Choose the [Debug→Go] menu option.

15.12.2 Running from reset

To reset your user system and run your program from the **Reset Vector** address, choose the [Debug→Reset Go] menu option or click the Reset Go toolbar button  . The program will run until it hits a breakpoint or a break condition is met. You can stop the program manually at any time by choosing the [Debug→Halt Program] menu option or by clicking the Halt Program toolbar button  .

Note:
The program will start running from whatever address is stored in the **Reset Vector** location. Therefore it is important to make sure that this location contains the address of your startup code.

15.12.3 Running to cursor

Sometimes as you are going through your application you may only want to run a small section of code, that would require many single steps to execute. In this case it would be useful to be able to run to a particular point. You can do this by using the **Go To Cursor** feature.

How to use the Go To Cursor feature:

1. Make sure that a **Source** or **Disassembly** view is open, showing the address at which you wish to stop.
2. Position the text cursor on the address at which you wish to stop by either clicking in the **Address** field or using the cursor keys.
3. Choose the **Go To Cursor** option from the pop-up menu.

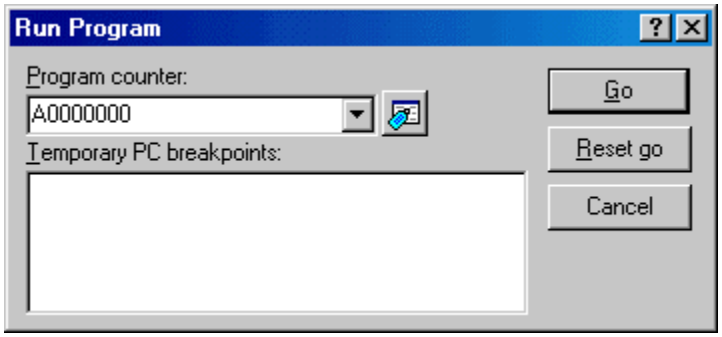
The debugging platform will run your code from the current PC value until it reaches the address indicated by the cursor's position.

Notes:

- If your program never executes the code at this address, the program will not stop. If this happens, you can stop code execution by pressing ESC, choosing the [Debug→Halt Program] menu option, or clicking the Halt Program toolbar button  .
- The **Go To Cursor** feature requires a temporary breakpoint – if you have already used all those available then the feature will not work, and the menu option will be disabled.

15.12.4 Running from a Specified Address

The [Run Program] dialog box allows the user to run the program from any address. Choose [Debug->Run...] to open the [Run Program] dialog box.



The following execution conditions can be specified in this dialog box:

[Program Counter]

Instruction address to start execution. The initial value is the current PC value.

[Temporary PC Breakpoints]

A temporary PC breakpoint. When execution started by this dialog box stops, this breakpoint is cleared.

Note: The [Temporary PC Breakpoints] feature requires a PC breakpoint - if you have already used all those available then the feature will not work.

Clicking the [Go] button starts execution according to the settings. Clicking the [Reset Go] button starts execution from the reset vector. Clicking the [Cancel] button closes this dialog box without executing instructions.

15.12.5 Single step

When you are debugging your code it is very useful to be able to step a single line or instruction at a time and examine the effect of that instruction on the system. In the **Source** view, a step operation will step a single source line. In the **Disassembly** view, a step operation will step a single assembly-language instruction. If the instruction calls another function or subroutine, you have the option to either step into or step over the function. If the instruction does not perform a call, then either option will cause the debugger to execute the instruction and stop at the next instruction.

If you choose to step into the function, the debugger will execute the call and stop at the first line or instruction of the function.

To step into the function:

- Click the Step In toolbar button  , **OR**
- Choose the [Debug→Step In] menu option.

If you choose to step over the function the debugger will execute the call and all of the code in the function (and any function calls that function may make) and stop at the next line or instruction of the calling function.

To step over the function:

- Click the Step Over toolbar button  , **OR**
- Choose the [Debug→Step Over] menu option.

During debugging, there are occasions when you may have entered a function, finished stepping through the instructions that you want to examine and would like to return to the calling function without tediously stepping through all the remaining code in the function. Alternatively you may have stepped into a function by accident, when you meant to step over it and so want to return to the calling function without stepping all the way through the current function. You can do this with the **Step Out** feature.

To step out of the current function:

- Click the Step Out toolbar button  , **OR**
- Choose the [Debug→Step Out] menu option.

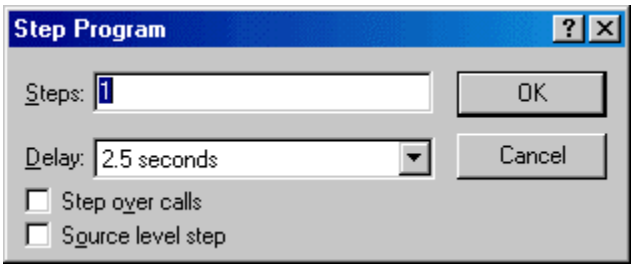
To choose the step mode to use while stepping:

Choose the [Debug→Step Mode] menu option.

Sub-menu	Function
Auto	Automatically chooses the step mode
Assembly	Steps through assembly instructions
Source	Steps through source code

15.12.6 Multiple steps

Sometimes you may find it useful to step through several instructions at a time. You can do this by using the **Step Program** dialog box. The dialog box also provides an automated step with a selectable delay between steps. Open it by choosing the [Debug→Step...] menu option. The **Step Program** dialog box is displayed.



[Steps]	Number of steps to be executed.
[Delay]	Delay between steps when the program is automatically stepped. [No Refresh] (Prevents the update of the windows) or 0 to 3 seconds can be selected in 0.5 second units.
[Step Over Calls]	Selecting this box steps over function calls.
[Source Level Step]	Selecting this box steps the program at the source level.

Click the **OK** button or press ENTER to start stepping.

15.13 Stopping your program

This section describes how you can halt execution of your application's code. This section describes how to do this directly by using the [Halt] button and by setting breakpoints at specific locations in your code.

15.13.1 Halting execution

When your program is running, the Halt Program toolbar button is enabled  , and when the program has stopped it is disabled  .

To stop the program:

- Click on the Halt Program toolbar button  , **OR**
- Choose the [Debug→Halt Program] menu option.

Your program's execution is halted, with the message ‘Break = User Break’ displayed on the **Status Bar**. The HEW will then update any open views. The cause of the last break can also be viewed in the **Platform** tab of the **Output** view.

15.13.2 Standard PC breakpoints

When you are trying to debug your program, you will want to be able to stop the program running when it reaches specific points in your code. You can do this by setting a program counter (PC) breakpoint on the line or instruction at which to want the execution to stop. The following instructions will show you how to quickly set and clear simple PC breakpoints. If more complex breakpoint operation is supported by the target, it may provide a **Breakpoints** view.

To set a program counter (PC) breakpoint:

1. Make sure that the **Disassemble** or **Source** view is open at the place at which you want to set a PC breakpoint.
2. Select the **Toggle Breakpoint** pop-up menu item, or press F9, at the line showing the address at which you want the program to stop.
3. You will see a red circle ● appear in the gutter to indicate that a PC breakpoint has been set.
4. It is also possible to add, remove and edit the current breakpoint setup by selecting the [Edit→Breakpoints] menu option.
5. Now when you run your program and it reaches the address at which you set the PC breakpoint, execution halts with the message 'PC Breakpoint' displayed in the **Debug** tab, and the **Source** or **Disassembly** view is updated with the PC breakpoint line highlighted.

Note:

The line or instruction at which you set a PC breakpoint is not actually executed – the program stops just before it is about to execute it. If you choose to **Go** or **Step** after stopping at the PC breakpoint, then the highlighted line will be the next instruction to be executed. When multiple targets are debugged, it is possible to specify either or both of them is stopped. For details, refer to section 15.17, Synchronizing Multiple Debugging Platforms.

To change the PC breakpoint setting by using the [Breakpoints] dialog box :

The breakpoint dialog can be displayed by selecting the [Edit→Breakpoints] menu option. It allows you to view the current breakpoints set in the workspace and view the code associated with each one. From this dialog it is also possible to remove one or all breakpoints.

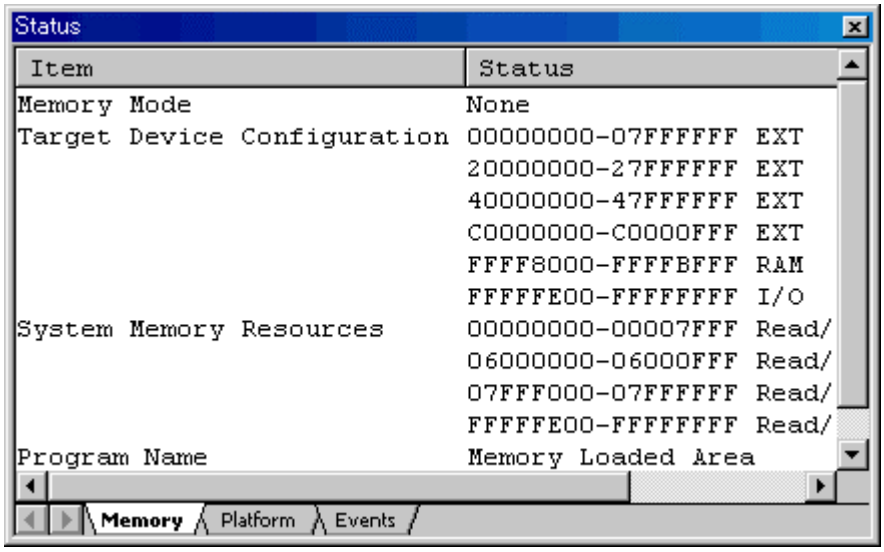
To toggle PC breakpoints:

It is possible to toggle PC Breakpoints either by double-clicking in the breakpoint (BP) column of the line at which the PC breakpoint is set, or by placing the caret on the line and using the F9 key. The display will cycle through the available standard breakpoint types – a red circle ● will be shown in the gutter.

15.14 Status window

To check the configuration and status of the debugging platform in the **Status** view, click the View Status button , or choose the [View→Status] menu item.

Configuration of Status window



The **Status** panel has three tabs:

Platform

Contains information about the current status of the debugging platform, typically including CPU type and mode, run status and timing information.

Memory

Contains information about the current memory status including the memory mapping resources and the areas used by the currently loaded object file.

Events

Contains information about the current event (breakpoint) status, including resource information.

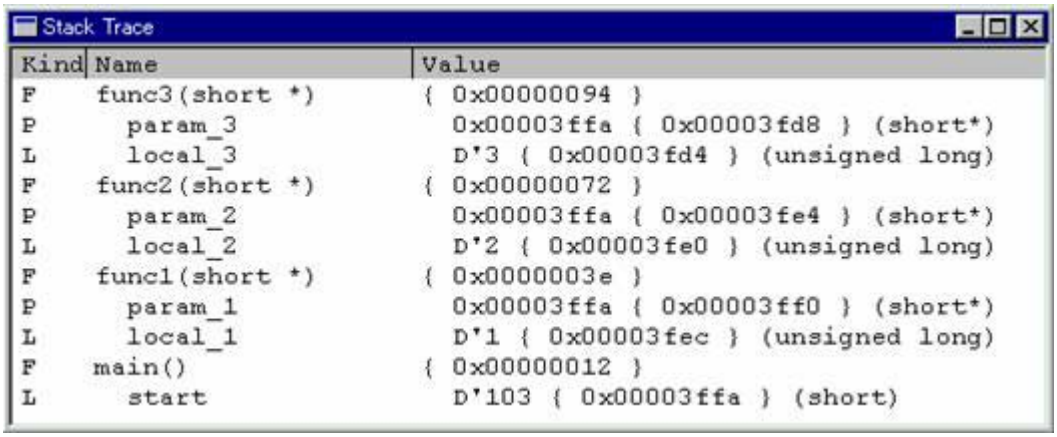
15.15 Viewing the function call history

The [Stack Trace] window shows the function call history.

15.15.1 Opening the Stack Trace Window

To open the [Stack Trace] window, choose [View->Code->Stack Trace] or click the [Stack Trace] toolbar button .

Configuration of Stack Trace window



The following items are displayed.

[Kind]	Indicates the type of the symbol. F: Function P: Function parameter ***1** L: Local variable ***1**
[Name]	Indicates the symbol name.
[Value]	Indicates the value, address, and type of the symbol.

***1** : Support for this function depends on the debugging platform.

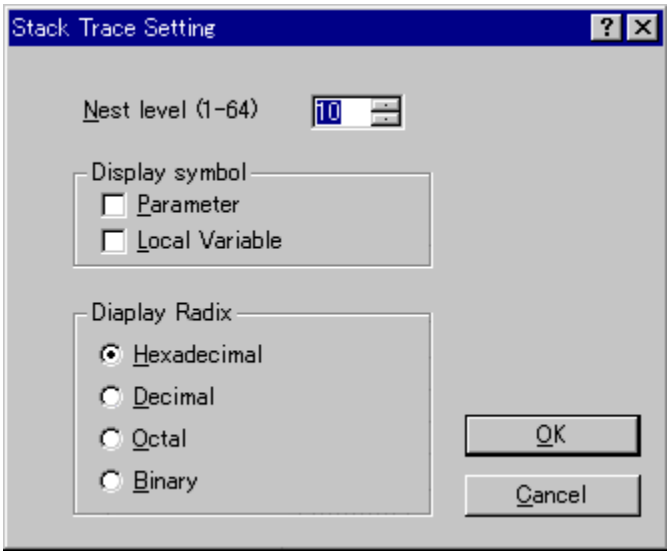
Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

Pop-up menu options	Function
Go to Source	Go to the associated source line.
View Setting...	Specifying the [Stack Trace] window settings.
Copy	Places a copy of the highlighted text into the Windows® clipboard.

15.15.2 Specifying the View

Choose [View Setting...] from the pop-up menu to open the [Stack Trace Setting] dialog box, which allows the user to specify the [Stack Trace] window settings.



[Nest level]	Specifies the level of function call nesting to be displayed in the [Stack Trace] window.
[Display symbol]	Specifies the symbol types to be displayed in addition to functions.
[Display Radix]	Specifies the radix for displays in the [Stack Trace] window.

15.15.3 Viewing the Source Program

Select a function and choose [Go to Source] from the pop-up menu to display, then the source program corresponding to the function, which has been selected by opening the [Editor] window, is displayed.

15.16 Using an external debugger

The High-performance Embedded Workshop can launch an external debugger tool. If you want to use another debugger then you must add it to the Tools menu.

The **Debugger** tab of the **Setup Customize** dialog is where the external debugger related information is configured. You may wish to use an older version of the debugger if certain targets are not currently supported in the new environment. Invoke it by selecting the [Setup→Customize...] menu option and then selecting the **Debugger** tab.

The first choice to make is which debug tool you would like to use.

Once this has been selected the external debugger must be configured.

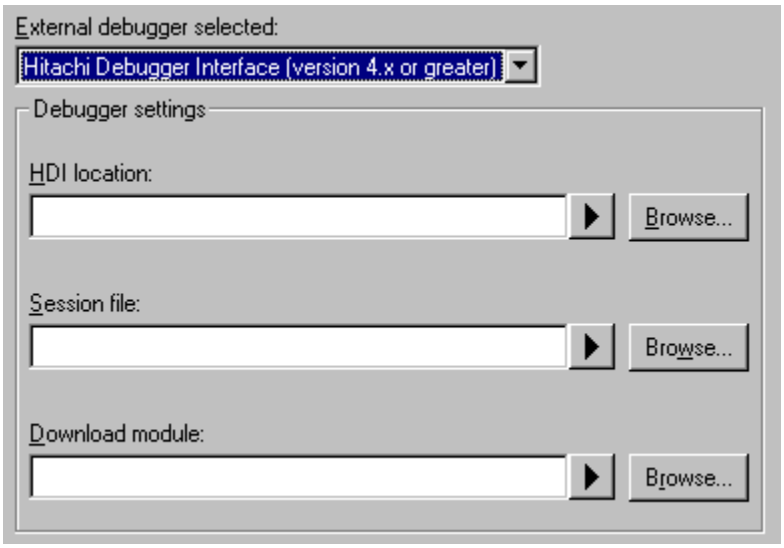
[Hitachi Debugger Interface (version 4.x or greater)	Configuring the Hitachi Debugging Interface to integrate with HEW
[Renesas PD debugger]	Configuring the PD debugger to integrate with HEW
[Other external debugger]	Configuring an external debugger to integrate with HEW
[Non selected]	Not use the external debugger

Click the Launch External Debugger toolbar button  to invoke the debugger with the specified session file.

After a build, if the download module has been updated, the HEW will switch back to the debugger to enable immediate debugging. Whilst using an external debugger, double-clicking in any source window will switch back to the HEW with the source file open at the line that was double-clicked.

15.16.1 Configuring the Hitachi Debugging Interface to integrate with HEW

The following details the information required to setup the Hitachi Debugging Interface to integrate with HEW and launch from the external debugger option in HEW.

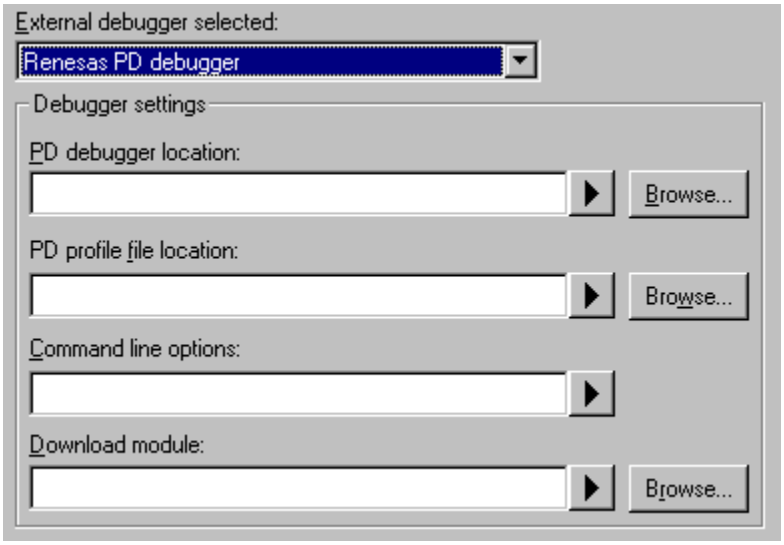


To configuring the Hitachi Debugging Interface to integrate with HEW:

1. Firstly, the location of the debugger executable must be specified. This must be version Hitachi Debugging Interface 4.0 or greater, otherwise its behaviour is not guaranteed. This may have been configured by the installation program or a project generation utility.
2. The second item of data is the session file. This tells the debugger which session to load when it is launched.
3. Finally, the location of the download module is required. This allows the HEW to automatically switch to the debugger when the download module changes after a build.

15.16.2 Configuring the PD debugger to integrate with HEW

The following details the information required to setup the PD debugger to integrate with HEW and launch from the external debugger option in HEW.

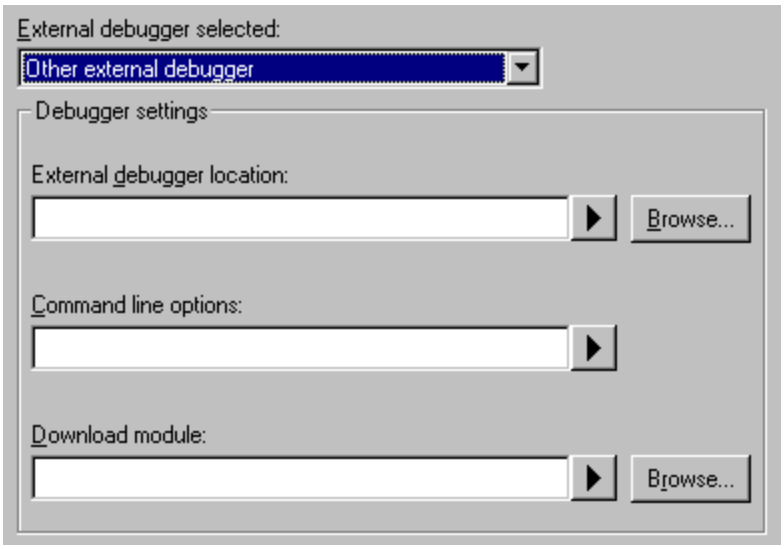


To configure the PD debugger to integrate with HEW:

1. Firstly, the location of the debugger executable must be specified. This may have been configured by the installation program or a project generation utility.
2. The second item of data is the profile file. This tells the debugger which profile file to load when it is launched. This file stores the debug setup information.
3. The third item of data are the command line options. This field allows additional options to be specified which can modify the behaviour of the external debugger.
4. Finally, the location of the download module is required. This allows the HEW to automatically switch to the debugger when the download module changes after a build.

15.16.3 Configuring an external debugger to integrate with HEW

The following details the information required to setup an external debugger which is not Hitachi Debugging Interface or the PD debugger to integrate with HEW and launch from the external debugger option in HEW.



To configure an external debugger to integrate with HEW:

1. Firstly, the location of the debugger executable must be specified. This may have been configured by the installation program or a project generation utility.
2. The second item of data are the command line options. This field allows additional options to be specified which can modify the behaviour of the external debugger.
3. Finally, the location of the download module is required. This allows the HEW to automatically switch to the debugger when the download module changes after a build.

15.17 Synchronizing multiple debugging platforms

Multiple debugging platforms can be operated at the same time in the HEW. There are two methods available to achieve this. These are outlined below; the external method was available in HEW 2.x. The Internal synchronization of debugger targets is the new preferred method for multiple target debugging in HEW 3.1.

15.17.1 External HEW synchronization

Initiating a HEW from another HEW synchronizes multiple debugging platforms. The HEW that initiates another HEW is called the master, and the initiated HEW is called the slave. Choose [Tools->Launch Slave HEW...] or click the [Launch Slave HEW] toolbar button  to initiate a slave HEW.

The slave HEW has the same functionality as the master HEW.

The slave HEW is notified of the following actions done in the master HEW to ensure synchronization of the slave HEW and the master HEW.

- Reset go
- Go
- Stop debugging

Note:

The master HEW can initiate multiple slave HEW applications, but slave HEW applications cannot be nested (no slave HEW can initiate another slave HEW).

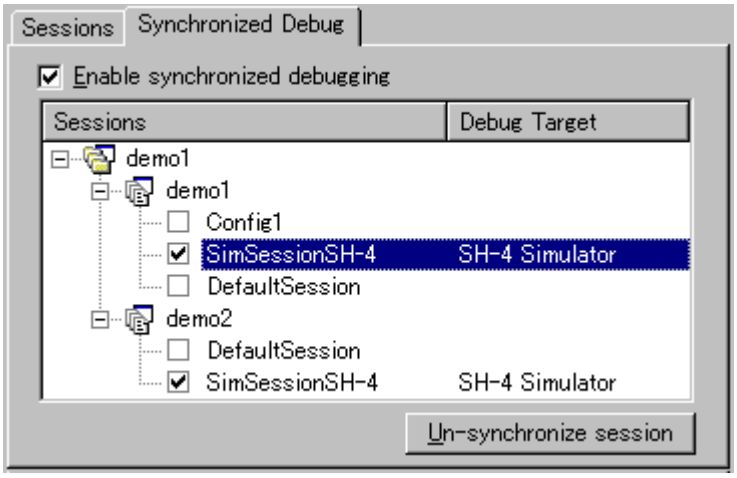
15.17.2 Internal HEW synchronization

The HEW also supports internal multiple target debugging. This will allow you to connect to multiple target components in the same HEW application. These targets can then be debugged simultaneously. The system allows the user to setup a number of sessions with different targets. Then when debugging the sessions can be synchronized so that certain events in one session can trigger the same events in the others. This is very similar to that seen in the External HEW synchronization section above. This facility though has the added advantage that it is easy to swap sessions and see what is happening in the same application.

To setup internal HEW synchronization:

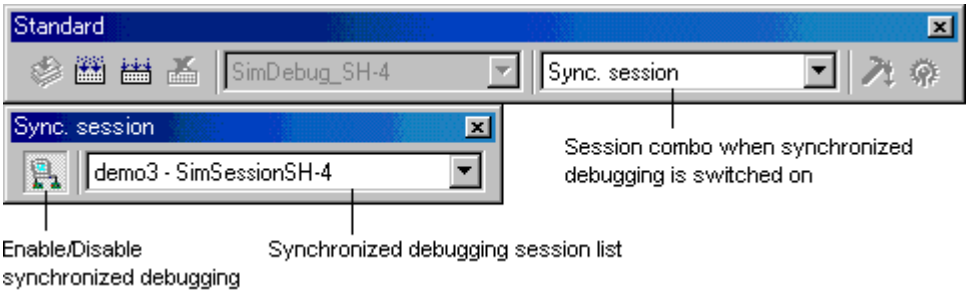
1. Select the [Debug->Debug Sessions…] menu item. The [Debug Sessions] dialog displayed.
2. Select the Synchronized Debug tab of the dialog.
3. Select the sessions you wish to synchronize. All currently available sessions in the workspace are displayed.

4. Click the synchronize session button. The icons for these sessions should changed to checked rather than unchecked.
5. Click the Enable synchronized debugging check box to switch this facility on.
6. Click OK to verify the changes.



To use internal HEW synchronization:

1. Follow the options as specified in to setup internal HEW synchronization.
2. Click on the session combo-box located on the standard toolbar. Select the Synch. Debug selection. This option is only available when synchronized debugging has been added to the system.
3. Once selected the HEW debugger enables the Synchronized debugging facilities. This means the addition of another toolbar this is named Sync. session.
4. The enable disable toolbar button on the Synch. session toolbar allows you to temporarily switch off the synchronization without losing your settings for this feature.
5. When enabled changing the session in the Sync. session combo box changes the session you are currently viewing. In the normal HEW debugging state this would mean the session is closed. In Synch. session you can have multiple sessions open and the currently selected one on this toolbar is the session you are viewing.
6. This system allows you to debug multiple targets or CPU cores simultaneously. Changing the session changes the views you can see on the screen and the data displayed on these views.



Note:

There are a number of capabilities that are synchronized when this facility is enabled.

The following tables display the capabilities when synchronized debugging is switched on. The example shows what happens in the two synchronized sessions.

Debugging function	Target Debugger Session 1	Target Debugger Session 2
User click “GO” in any Session	“GO”	“GO”
User click “STEP Into/out/over” in any Session	“Step”	“Step”
User click “ESC” in any Session	“BREAK”	“BREAK”
-	“BREAK” by breakpoints, illegal access or due to illegal User Program	Stop running (Same Effect of pressing ESC)
-	Stop running (Same Effect of pressing ESC)	“BREAK” by breakpoints, illegal access or due to illegal User Program
RESET CPU in any Session	RESET CPU	RESET CPU

The following tables display the capabilities when synchronized debugging is switched off. The example shows what happens in the two synchronized sessions.

Debugging function	Target Debugger Session 1	Target Debugger Session 2
User click “GO” in Session 1	“GO”	No activities, “GO” can only be executed manually by User
User click “GO” in Session 2	No activities, “GO” manually executed by User	“GO”
User click “STEP” in session 1	“STEP”	No activities, must “STEP” manually by User
User click “STEP” in session 2	No activities, must “STEP” manually by User	“STEP”
User click “ESC” in session 1	“BREAK”	Still executing, if previously executing User Target Program
User click “ESC” in session 2	Still executing, if previously executing User Target Program	“BREAK”
-	“BREAK” by breakpoints, illegal access or due to illegal User Program	Still executing, if previously executing User Target Program
-	Still executing, if previously executing User Target Program	“BREAK” by breakpoints, illegal access or due to illegal User Program
RESET CPU in session 1	RESET CPU	No activities
RESET CPU in session 2	No activities	RESET CPU

Note:

Another difference to the standard debugging system is that it is possible to view the download modules in the workspace window for all synchronized sessions. This allows modules to be downloaded for any of the available sessions easily.

15.18 Existing HEW functions dependent on the debugger

15.18.1 Looking at labels

In addition to the debugging information that the HEW uses to link your program's source code to the actual code in memory, the **debug object file** also contains symbolic information. This is a table of text names that represent an address in the program and are referred to as **labels** in HEW. In the **Disassembly** view, you will see the first eight characters of the label in place of the corresponding address, and as part of an instruction's operand.

Note:

An instruction's operand is replaced with a label name if the operand and label value match. If two or more labels have the same value, then the label that comes first alphabetically will be displayed.

(1) Listing labels

To see a list of all the labels defined in the current debugger session, click the View Labels toolbar button  , or select the [View–>Symbol–>Labels] menu option.

Configuration of Label window

- You can view symbols, sorted either alphabetically (by ASCII code) or by address value, by clicking on the respective column heading.
- You can quickly toggle a software break at the entry point of a function by double clicking in the BP (breakpoint) column. Alternatively, right-click to show the pop-up menu and select Break.

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of [Toolbar display] and [Customize toolbar...] are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options	Toolbar bottom	Function
Add...		Adding a label.
Edit...		Editing a label.
Delete		Deleting a label.
Delete All		Deleting all labels.
Load...		Loading labels from a file.
Save		Saving labels into a file.
Save As...		Saving labels into a file.
Find...		Searching for a label.
Fine Next		Searching for the next label.
View Source		Viewing the source corresponding to a label.
Print		Prints the currently displayed contents.
Toolbar display	-	Showing/hiding toolbar buttons.
Customize toolbar...	-	Customizing toolbar buttons.

(2) Adding a Label

Choose [Add..] from the pop-up menu and open the [Add Label] dialog box to add a label:

Enter the new label name into the [Name] field and the corresponding value into the [Address] field and press [OK]. The [Add Label] dialog box closes and the label list is updated to show the new label. When an overloaded function or a class name is entered in the [Address] field, the [Select Function] dialog box opens for you to select a function. For details, refer to section 15.18.2 (1) Supporting duplicate labels.

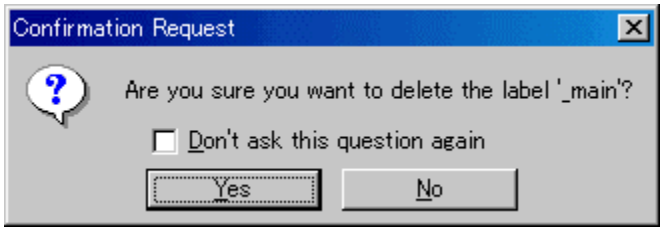
(3) Editing a Label

Choose [Edit...] from the pop-up menu and open the [Edit Label] dialog box to edit a label:

Edit the label name and value as required and then press [OK] to save the modified version in the label list. The list display is updated to show the new label details. When an overloaded function or a class name is entered in the [Address] field, the [Select Function] dialog box opens for you to select a function. For details, refer to section 15.18.2 (1) Supporting duplicate labels.

(4) Deleting a Label

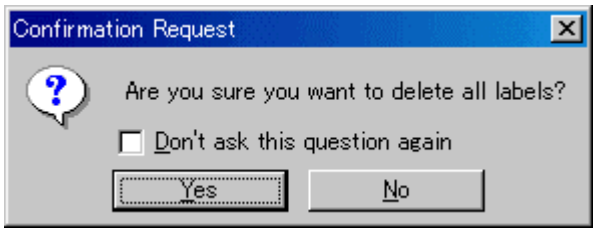
To delete a label, select the label and choose [Delete] from the pop-up menu. A confirmation message box appears:



If you click [OK], the label is removed from the list and the window display is updated. If the message box is not required then do not select the [Delete Label] option of the [Confirmation] sheet in the HEW [Options] dialog box.

(5) Deleting All Labels

To delete all the labels from the list, choose [Delete All] from the pop-up menu. A confirmation message box appears:



If you click [OK], all the labels are removed from the HEW system's symbol table and the list display will be cleared. If the message box is not required, do not select the [Delete All Labels] option of the [Confirmation] sheet in the HEW [Options] dialog box.

(6) Loading Labels from a File

A symbol file can be loaded and merged into the HEW's current symbol table. Choose [Load...] from the pop-up menu to open the [Load Symbols] dialog box:

The dialog box operates like a standard Windows® [Open file] dialog box; select the file and click [Open] to start loading. The standard file extension for symbol files is ".sym". When the symbol loading is complete a confirmation message box may be displayed showing how many symbols have been loaded (this can be switched off in the [Confirmation] sheet on the HEW [Options] dialog box).

(7) Saving Labels into a File

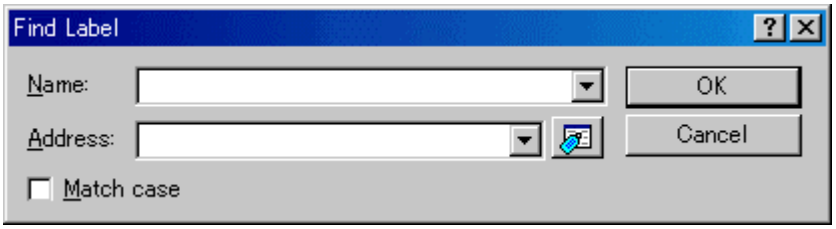
Choose [Save As...] from the pop-up menu to open the [Save Symbols] dialog box. The [Save Symbols] dialog box operates like a standard Windows® [Save File As] dialog box. Enter the name for the file in the [File name] field and click [Save] to save the HEW's current label list to a symbol file. The standard file extension for symbol files is ".sym".

See Reference 6, Symbol File Format, for symbol file format.

Once a file is specified by the [Save As...] menu, the current symbol table can be saved in the same symbol file just by choosing [Save] from the pop-up menu.

(8) Searching for a Label

Choose [Find...] from the pop-up menu to open the [Find Label] dialog box:



Enter the label name that you wish to find into the edit box and click [OK] or press the Enter key. The HEW searches the label list for a label name containing the text that you entered.

Note:

Only the label is stored by 1024 characters of the start, therefore the label name must not overlap mutually in 1024 characters or less. Labels are case sensitive.

(9) Searching for the Next Label

Choose [Find Next] from the pop-up menu to find the next occurrence of the label containing the text that you entered.

(10) Viewing the Source Corresponding to a Label

Select a label and choose [View Source] from the pop-up menu to open the [Source] or [Disassembly] window containing the address corresponding to the label.

15.18.2 Elf/Dwarf2 support

The HEW supports the Elf/Dwarf2 object file format for debugging applications written in C/C++ and assembly language for Renesas microcomputers. It provides a powerful way of accessing, observing and modifying the symbolic level debugging information about the user application that is running.

Key Features:

- Source level debugging
- C/C++ operators
- C/C++ expression (casting, pointers, references, etc.)
- Ambiguous function names
- Overlay memory loading
- Watch – locals and user defined
- Stack trace

(1) C/C++ operators

The C/C++ language operators are available:

```
+, -, *, /, &, |, ^, ~, !, >>, <<, %, (, ), <, >, <=, >=, ==, !=, &&, ||
Buffer_start + 0x1000
#R1 | B'10001101
((pointer + (2 * increment_size)) & H'FFFF0000) >> D'15
!(flag ^ #ER4)
```

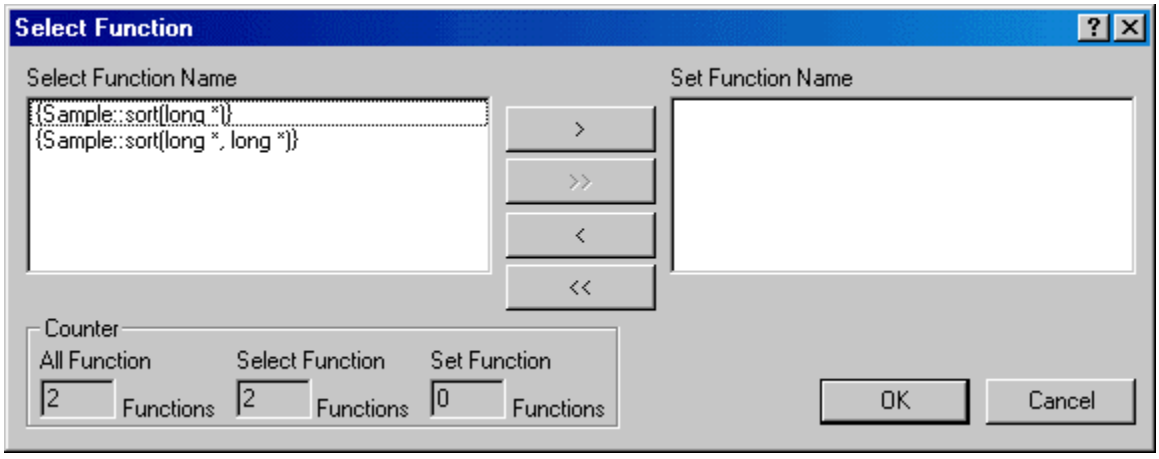
(2) C/C++ expressions

Expression examples:

Object.value	Specifies direct reference of a member (C/C++)
p_Object->value	Specifies indirect reference of a member (C/C++)
Class::value	Specifies reference of a member with class (C++)
*value	Specifies a pointer (C/C++)
&value	Specifies a reference (C/C++)
array[0]	Specifies an array (C/C++)
Object.*value	Specifies reference of a member with pointer (C++)
::g_value	Specifies reference of a global variable (C/C++)
Class::function(short)	Specifies a member function (C++)
(struct STR) *value	Specifies a cast operation (C/C++)

(3) Supporting duplicate labels

In some languages, for example in C++ overloaded functions, a label may represent more than one address. Just entering the label name is ambiguous, so the HEW will display the **Select Function** dialog box to display overloaded functions and member functions.



Select overloaded functions or member functions in the **Select Function** dialog box. Generally, only one function can be selected at one time (except for setting breakpoints, as multiple functions can be selected). This dialog box has three areas:

[Select Function Name] list box	Displays the same-name functions or member functions and their detailed information.	
[Set Function Name] list box	Displays the function to be set and their detailed information.	
[Counter group] edit box	All Function	Displays the number of same-name functions or member functions.
	Select Function	Displays the number of functions displayed in the **Select Function Name** list box.
	Set Function	Displays the number of functions displayed in the **Set Function Name** list box.

(a) Selecting a function

Click the function you wish to select in the **Select Function Name** list box, and click the **>** button. You will see the selected function in the **Set Function Name** list box. To select all functions in the **Select Function Name** list box, click the **>>** button.

(b) Deselecting a function

Click the function you wish to deselect from the **Set Function Name** list box, and click the **<** button. To deselect all functions, click the **<<**button. The deselected function(s) will be moved from the **Set Function Name** list box back to the **Select Function Name** list box.

(c) Setting a function

Click the **OK** button to set the functions displayed in the **Set Function Name** list box. The functions are set and the **Select Function** dialog box closes.

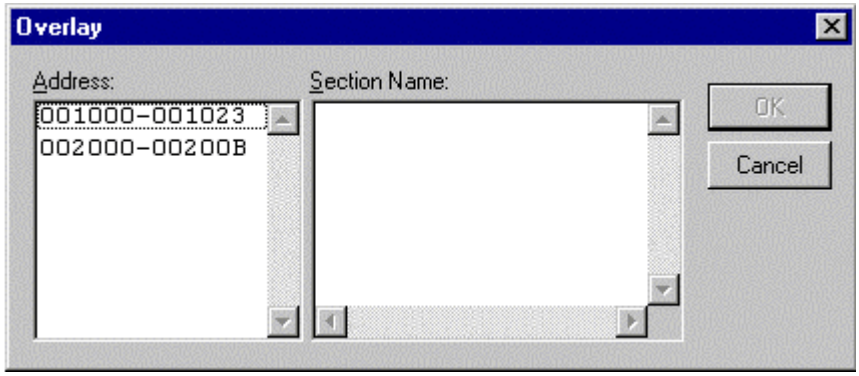
Click the **Cancel** button to close the dialog box without setting the functions.

(4) Debugging an Overlay Program

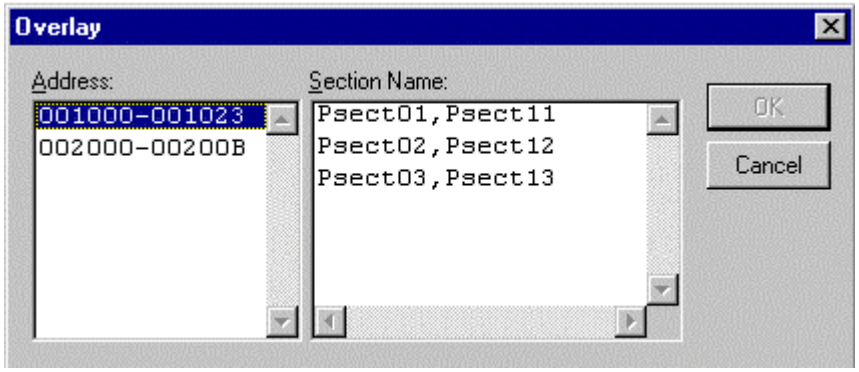
Programs making use of the **Overlay** function can be debugged. This section explains the settings for using the **Overlay** function.

(a) Displaying section group

When the **Overlay** function is used (i.e. when several section groups are assigned to the same address range), the address ranges and section groups are displayed in the **Overlay** dialog box.



Open the **Overlay** dialog box by choosing the [Debug→Overlay] menu option. This dialog box has two areas: the **Address** list box and the **Section Name** list box. The **Address** list box displays the address ranges used by the **Overlay** function. Click to select one of the address ranges in the Address list box.

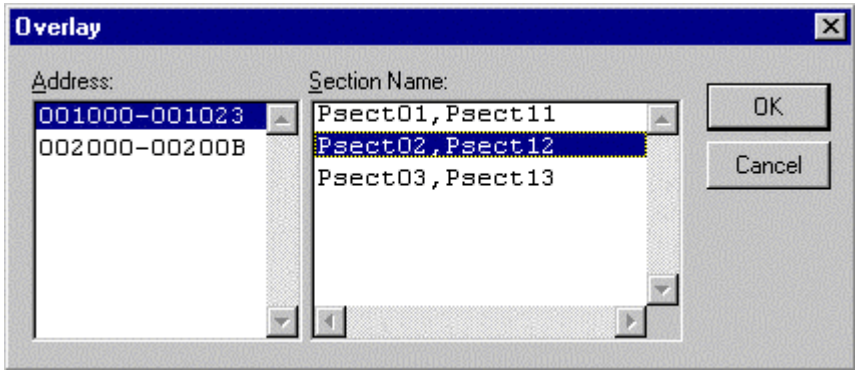


The **Section Name** list box displays the section groups assigned to the selected address range.

(b) Setting section group

When using the **Overlay** function, the highest-priority section group must be selected in the **Overlay** dialog box, otherwise the HEW will operate incorrectly.

Firstly, click one of the address ranges displayed in the **Address** list box. The section groups assigned to the selected address range will then be displayed in the **Section Name** list box. Click to select the section group with the highest-priority among the displayed section groups.



After selecting a section group, clicking the **OK** button stores the priority setting and closes the dialog box. Clicking the **Cancel** button closes the dialog box without storing the priority setting.

Note:

Within the address range used by the **Overlay** function, the debugging information for the section specified in the **Overlay** dialog box is referred to. Therefore, the same section of the currently loaded program must be selected in the **Overlay** dialog box.

15.18.3 Looking at variables

This section describes how you can look at variables in the source program.

(1) Tooltip Watch

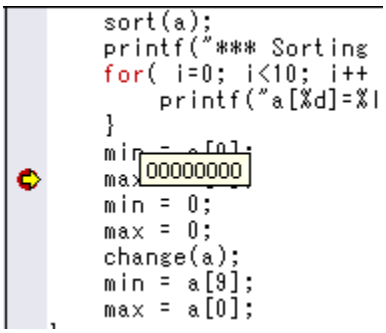
The quickest way to look at a variable in your program is to use the **Tooltip Watch** feature.

To use Tooltip Watch:

1. Select [Setup->Options…] menu option.
2. Select the [Editor] tab. The [Options] dialog box is displayed.
3. Check the [Enable tooltip watch] check box.
4. Click [OK].

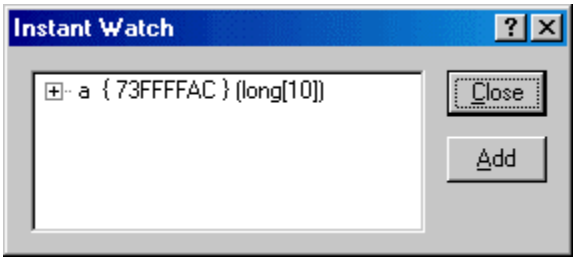
To view a tooltip watch on the Editor window:

1. Open the **Source** window showing the variable that you want to examine.
2. Rest the mouse cursor over the variable name that you want to examine. A tooltip will appear near the variable containing basic watch information for that variable.



(2) Instant Watch

Display the source file containing the variable that you want to examine on the [Editor] window.
Rest the mouse cursor over the variable name that you want to examine and choose [Instant Watch] from the pop-up menu; the [Instant Watch] dialog box will appear and display the variable at the cursor location.



"+" shown to the left of the variable name indicates that the information may be expanded by clicking on the variable name, and "-" indicates that the information may be collapsed. Clicking [Add] registers the variable in the [Watch] window. Clicking [Close] closes the window without registering the variable in the [Watch] window.

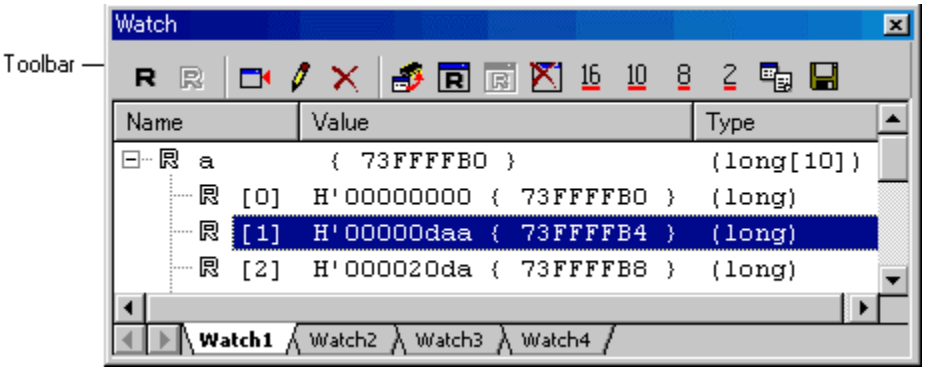
(3) Watch window

The HEW allows you to open Watch windows, which contain a list of variables and their values.

(a) Opening a Watch Window

To open a [Watch] window, click on the [Watch] toolbar button  or choose [View->Symbol->Watch] if it is visible. A [Watch] window opens. Initially the contents of the window will be blank.

Configuration of Watch window



- This window allows the user to view and modify C/C++-source level variables.
- The contents of this window are displayed only when the debugging information available in the absolute file (*.abs) includes the information on the C/C++ source program. The variable information is not displayed if the source program information is excluded from the debugging information during optimization by the compiler. In addition, the variables that are declared as macro cannot be displayed.
- A variable can be dragged from the [Editor] window and dropped into the [Watch] window.

The following items are displayed.

[Name]	Name of the variable.
[Value]	Value and assigned location. The assigned location is enclosed by { }.
[Type]	Type of the variable

The [R] mark shows that the value of the variable can be updated during user program execution. When the color of the [R] mark is black, a value is real-time updated.

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of [Toolbar display] and [Customize toolbar...] are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options		Toolbar bottom	Function
Auto Update			Marks the selected variable with a bold R and updates the variable in real time.
Auto Update All			Marks all variables with bold Rs and updates all variables in real time.
Delete Auto Update			Marks the selected variable with an outlined R and cancels realtime update.
Delete Auto Update All			Marks all variables with outlined Rs and cancels realtime update.
Add Watch...			Launches the Add Watch dialog box, allowing the user to enter a variable or expression to be watched.
Edit Value...			Launches the Edit Watch dialog box, allowing the user to change the variable's value.
Delete			Removes the variable indicated by the text cursor from the Watch window.
Delete All			Removes all the variables from the Watch window.
Radix	Hexadecimal		Display in hexadecimal.
	Decimal		Display in decimal.
	Octal		Display in octal.
	Binary		Display in binary.
Copy			Places a copy of the highlighted text into the Windows® clipboard.
Save As...			Saves the currently displayed contents.
Go To Memory			Opens a Memory window for the address.
Toolbar display		-	Showing/hiding toolbar buttons.
Customize toolbar...		-	Customizing toolbar buttons.

(b) Adding a Watch Item

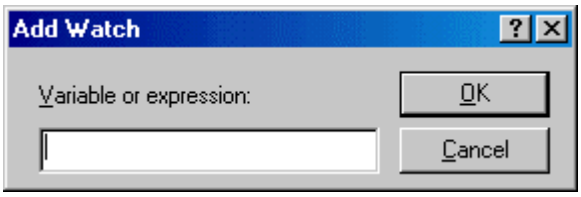
Use the [Add Watch] dialog box in the [Watch] window to add Watch items to the [Watch] window.

To use Add Watch from a [Watch] window:

Open the [Watch] window.

Choose [Add Watch] from the pop-up menu.

The [Add Watch] dialog box opens:



Enter the name of the variable that you wish to watch and click [OK]. The variable is added to the [Watch] window. A variable can be dragged from the [Editor] window and dropped into the [Watch] window.

Note:

If the variable that you have added is a local variable that is not currently in scope, the HEW will add it to the [Watch] window but its value will be 'Not available now'.

(c) Expanding a Watch Item

If a watch item is a pointer, array, or structure, then you will see a plus sign (+) expansion indicator to left of its name, this means that you can expand the watch item. To expand a watch item, click on it. The item expands to show the elements (in the case of structures and arrays) or data value (in the case of pointers) indented by one tab stop, and the plus sign changes to a minus sign (-). If the elements of the watch item also contain pointers, structures, or arrays then they will also have expansion indicators next to them.

To collapse an expanded watch item, click on the item again. The item's elements will collapse back to the single item and the minus sign changes back to a plus sign.

(d) Editing a watch item's value

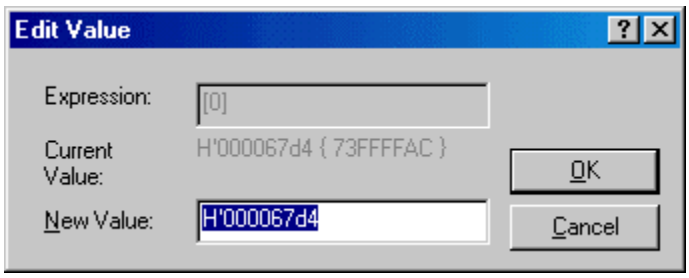
You may wish to change the value of a watch variable, e.g. for testing purposes or if the value is incorrect due to a bug in your program. To change a watch item's value use the **Edit Value** function.

To edit a watch item's value:

Enter a value directly in the window.

In another way, select the item to edit by clicking on it, you will see a flashing cursor on the item.
Choose [Edit Value] from the pop-up menu.

The [Edit Value] dialog box opens:



Enter the new value or expression in the [New Value] field and click [OK]. The [Watch] window is updated to show the new value.

(e) Specifying Realtime Update

The R mark shown to the left of each variable indicates whether the variable is updated in real time.

A pop-up menu containing the following options is available in the [Watch] window:

Auto Update	Marks the selected variable with a bold R and updates the variable in real time.
Auto Update All	Marks all variables with bold Rs and updates all variables in real time.
Delete Auto Update	Marks the selected variable with an outlined R and cancels realtime update.
Delete Auto Update All	Marks all variables with outlined Rs and cancels real-time update.

(f) Deleting a watch item

To delete a watch item, select it from the **Watch** view and choose the [Delete] option from the pop-up menu. The item is deleted and the **Watch** view is updated.

To delete all watch item, select it from the **Watch** view and choose the [Delete All] option from the pop-up menu. The all items are deleted and the **Watch** view is updated.

Watch items that you have set in the **Watch** window are saved in the session file.

(g) Modifying the Radix

The radix for the selected variable display can be modified by choosing [Radix] from the pop-up menu.

(h) Saving the Watch Window Contents in a File

To save the contents of the [Watch] window, choose [Save As...] from the pop-up menu; the Save As dialog box opens. It allows the user to specify the name of a file and to save the contents of the [Watch] window in the file. If the [Append] check box is selected, the window contents are appended to the existing file, and if it is not selected, the existing file is overwritten.

(i) Opening a Memory Window

The contents of the memory area to which the selected variable is assigned can be displayed in the [Memory] window. Choosing [Go To Memory...] from the pop-up menu opens the [Memory] window.

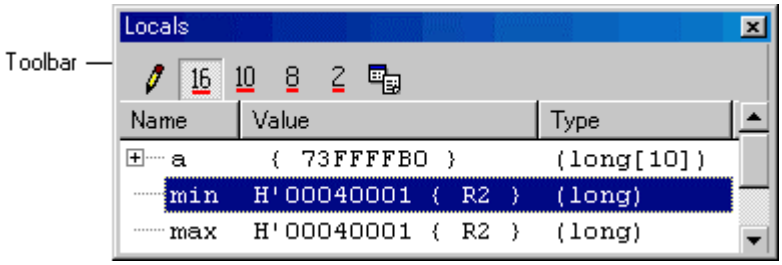
(4) Locals window

The local variables and their values can be displayed in the [Locals] window.

(a) Opening the Locals Window

To open the [Locals] window, click the [Locals] toolbar button  or choose [View->Symbol->Locals].

Configuration of Locals window



As you debug your program, the [Locals] window will be updated. If a local variable is not initialized when defined, then the value in the [Locals] window will be undefined until a value is assigned to the local variable.

The local variable values and the radix for local variable display can be modified in the same manner as in the [Watch] window.

Option

Clicking the right-hand mouse button displays a pop-up menu containing available options.

A basic operation is allocated to the toolbar.

The functions of [Toolbar display] and [Customize toolbar...] are also included in the pop-up menu displayed by right-clicking the toolbar area.

Pop-up menu options		Toolbar bottom	Function
Edit Value...			Launches a dialog box to modify the selected variable's value.
Radix	Hexadecimal		Display in hexadecimal.
	Decimal		Display in decimal.
	Octal		Display in octal.
	Binary		Display in binary.
Copy			Places a copy of the highlighted text into the Windows® clipboard.
Toolbar display		-	Showing/hiding toolbar buttons.
Customize toolbar...		-	Customizing toolbar buttons.

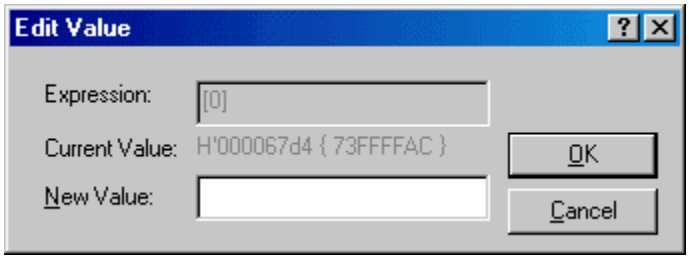
(b) Editing a local Item's Value

You may wish to change the value of a local variable, e.g. for testing purposes or if the value is incorrect due to a bug in your program. To change a watch item's value use the Edit Value function.

Editing a local item's value:

Enter a value directly in the window.

In another way, select the item to edit by clicking on it, you will see a flashing cursor on the item.
Choose [Edit Value] from the pop-up menu.
The [Edit Value] dialog box opens:



Enter the new value or expression in the [New Value] field and click [OK]. The [Locals] window is updated to show the new value.

(c) Modifying the Radix

The radix for the selected variable display can be modified by choosing [Radix] from the pop-up menu.

16. Technical Support

16.1 Check for Updates

To check for HEW product updates or service packs:

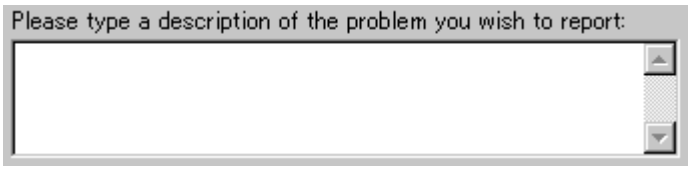
1. Select [Help->Technical support->Check web site for updates].
2. Your default web browser is invoked and defaults to the HEW download page for your region.
3. Browse this area for HEW updates to fix bugs or add new features.

16.2 Creating a Bug Report

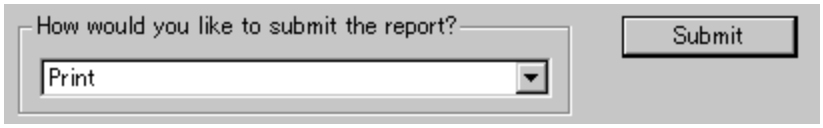
Occasionally you may experience some unforeseen problems with the HEW application. If a problem does occur that results in a application crash the HEW bug tracking program will be invoked automatically. This allows you to compile a bug report and this can then be sent to your technical support contact in a variety of ways. It is also possible to invoke this tracker program manually. This is described below:

To create and send a HEW bug report:

1. Select [Help->Technical support->Create a HEW bug report...].
2. Detailed information is generated from your HEW system. This may take some time. The [Submit a Bug Report] dialog is then displayed.
3. It is possible to then add additional information concerning the exact issue you have found in the large edit box.



4. Once you are happy with your report, you can choose the method of sending the report in the [How would you like to submit the report?] drop list. You can print it, e-mail it, or save it to a disk.



5. Then click [Submit]. This will send the report.

Reference

1. List of Menus

1.1 List of File Menu

Menu	Menu Option	Keyboard Shortcuts	Toolbar Button	Function
File	New	Ctrl+N		Creates a new document.
	Open...	Ctrl+O		Opens an existing document.
	Close	Ctrl+F4	-	Closes the active document.
	New Workspace...	-	-	Creates a new workspace.
	Open Workspace...	-	-	Opens an existing workspace.
	Save Workspace	-	-	Saves the current workspace.
	Close Workspace	-	-	Closes the current workspace.
	New Session...	-	-	Creates a new session.
	Import Session...	-	-	Import an existing session.
	Save Session	-	-	Save the current session.
	Save Session As...	-	-	Save the current session with a new session name.
	Refresh Session	-	-	Reload the session file for the current session.
	Save	Ctrl+S		Save the active document.
	Save All	Ctrl+Shift+S		Save all modified documents in the workspace.
	Save As...	-	-	Save the active document with a new file name.
	Page Setup...	-	-	Change the printing options.
	Print...	Ctrl+P		Prints the active document.
	Recent Files	-	-	Open this document.
	Recent Workspaces	-	-	Open this workspace.
	Exit	-	-	Exit HEW.

1.2 List of Edit Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Function
Edit	Undo		Ctrl+Z	-	Reverses the last editing operation.
	Redo		Ctrl+Y	-	Repeats the last undone editing operation.
	Cut		Ctrl+X		Removes highlighted text and places it on the Windows® clipboard.
	Copy		Ctrl+C		Places a copy of the highlighted text into the Windows® clipboard.
	Paste		Ctrl+V		Copies the contents of the Windows® clipboard into the active window at the position of the insertion cursor.
	Clear		Delete	-	Removes highlighted text (it is not copied to the Windows® clipboard)
	Select All		Ctrl+A	-	Selects (i.e. highlights) the entire contents of the active window
	Find...		Ctrl+F		Find text in the current file.
	Find In Files...		F4		Find text in multiple files.
	Replace...		Ctrl+H	-	Replace text in the current file.
	Goto Line...		Ctrl+G	-	Jumps to a line in a file.
	Match Braces		Shift+Ctrl+M		Finds a matching brace.
	Bookmarks	Toggle Bookmark	Ctrl+F2		Sets a bookmark at the current line or clears a bookmark at the current line.
		Next Bookmark	F2		Jumps to the next bookmark in the current file from the current line.
		Previous Bookmark	Shift+F2		Jumps to the previous bookmark in the current file from the current line.
		Clear All Bookmarks	-		Clears all bookmarks in the current file.
	Templates	Define Templates...	-		Defines a template.
		Insert Template...	Shift+Ctrl+T		Inserts a template.
	Toggle Breakpoint		F9		Sets or clears a software breakpoint at the line showing the address.
	Enable/Disable Breakpoint		Ctrl+F9		Enable or disable the current software breakpoint.
	Define Column Format...		-	-	Set the status of editor columns.
	Source Breakpoints...		Ctrl+B	-	Opens the [Breakpoins] dialog box.
	Evaluate...		-	-	Evaluates simple and complex expressions.

1.3 List of View Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Function
View	Differences		-	-	Opens the [Difference] window.
	Map... *1		-		Opens the [Map] window.
	Command Line		Ctrl+L		Opens the [Command Line] window.
	TCL Toolkit		Shift+Ctrl+K		See the "Tcl/Tk Additional document".
	Workspace		Alt+K		Opens the [Workspace] window.
	Output		Alt+U		Opens the [Output] window.
	Disassembly		Ctrl+D		Opens the [Disassembly] window.
	CPU	Registers	Ctrl+R		Opens the [Registers] window.
		Memory	Ctrl+M		Opens the [Memory] window.
		IO	Ctrl+I		Opens the [IO] window.
		Status	Ctrl+U		Opens the [Status] window.
	Graphic	Image...	Shift+Ctrl+G		Opens the [Image] window.
		Waveform...	Shift+Ctrl+V		Opens the [Waveform] window.
	Code	Stack Trace	Ctrl+K		Opens the [Stack Trace] window.

***1 :** Support for this function depends on the debugging platform.

1.4 List of Project Menu

Menu	Menu Option	Keyboard Shortcuts	Toolbar Button	Function
Project	Set Current Project	-	-	Set this project as the current project.
	Insert Project...	-	-	Add project to workspace.
	Dependent Projects...	-	-	Show dependent projects.
	Edit Project Configuration...	-	-	Edit the project configuration.
	Create Project Type...	-	-	Create a new project type.
	Add Files...	-	-	Add file(s) to project.
	Remove Files...	-	-	Remove file(s) from project.
	File Extensions...	-	-	Display current project file extensions.
	Components...	-	-	Load/unload components.

1.5 List of Build Menu

Menu	Menu Option	Keyboard Shortcuts	Toolbar Button	Function
Build	Toolchain...	-	-	Sets options for a build phase.
	Build File	-		Build the selected files.
	Build	-		Build out of date project files.
	Build All	-		Build project files, regardless of whether the project files are out o date.
	Build Multiple...	-	-	Build multiple projects.
	Update All Dependencies	-	-	Update a project's dependencies.
	Stop Build	Ctrl+Break		Stop a build.
	Terminate Current Tool	-	-	Forcibly terminate a task.
	Build Phase...	-	-	Add, remove and modify phase.
	Build Configurations...	-	-	Select the current configuration.
	Linkage Order...	-	-	Customize the HEW linkage order.
	Generate Makefile...	-	-	Generate a makefile.

Available only when there is a toolchain installed.

1.6 List of Debug Menu

Menu	Menu Option	Keyboard Shortcuts	Toolbar Button	Function
Debug	Debug Sessions...	-	-	Opens the [Debug Sessions] dialog box to list, add, or remove the debug session.
	Debug Settings...	-	-	Opens the [Debug Settings] dialog box to set the debugging conditions or download modules.
	Reset CPU	-		Resets the microprocessor.
	Go	F5		Starts executing the user program at the current PC.
	Reset Go	Shift+F5		Executes the user program from the reset vector address.
	Go to Cursor	-		Starts executing the user program at the current PC and continues until the PC equals the address indicated by the current text cursor position.
	Set PC to Cursor	-		Changes the value of the Program Counter (PC) to the address at the row of the text cursor.
	Run...	-	-	Launches the [Run Program] dialog box allowing the user to enter temporary breakpoints before executing the user program.
	Display PC	Ctrl+Shift+Y		Opens the [Editor] or [Disassembly] window at the address of the PC.
	Step In	F11		Executes a block of user program before breaking.
	Step Over	F10		Executes a block of user program before breaking. If a subroutine call is reached, then the subroutine will not be entered.
	Step Out	Shift+F11		Executes sufficient user program to reach the end of the current function.
	Step...	-	-	Launches the [Program Step] dialog box allowing the user to modify the settings for stepping.
	Step Mode Auto	-	-	Steps only one source line when the [Editor] window is active. When the [Disassembly] window is active, stepping is executed in a unit of assembly instructions.
	Assembly	-	-	Executes stepping in a unit of assembly instructions.
	Source	-	-	Steps only one source line.
	Halt Program	-		Stops the execution of the user program.
	Initialize	-	-	Disconnects the debugging platform and connects it again.
	Connect ***1**	-		Connects the debugging platform.
	Disconnect ***1**	-		Disconnects the debugging platform.
	Save Memory...	-	-	Saves the specified memory area data to a file.
	Verify Memory... ***1**	-	-	Verifies file contents against memory contents.
	Download Modules	-	-	Downloads the object program.
	Unload Modules	-	-	Unloads the object program.

***1 :** Support for this function depends on the debugging platform.

1.7 List of Setup Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Function
Setup	Customize...		-	-	Customize the HEW application.
	Options...		-	-	Sets option of the HEW application.
	Format Views...		-	-	Configure fonts, colors, keywords and so on, for the window.
	Radix	Hex	-		Sets radix to base Hex.
		Decimal	-		Sets radix to base Decimal.
		Oct	-		Sets radix to base Octal.
		Bin *1	-		Sets radix to base Binary.

***1 :** Support for this function depends on the debugging platform.

1.8 List of Tools Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Function
Tools	Administration...		-	-	Control the components.
	Change Toolchain Version...		-	-	Change toolchain version.
	Version Control	Select...	-	-	Select a version control system.
		Configure...	-	-	Setup the version control system.
		Add File(s)	-		Add files to Visual SourceSafe.
		Remove File(s)	-		Get a read-only copy of a file from Visual SourceSafe.
		Get File(s)	-		Get a read-only copy of a file from Visual SourceSafe.
		Check Out File(s)	-		Check out a writable copy of a file from Visual SourceSafe.
		Check In File(s)	-		Check in your edits to a file into Visual SourceSafe.
		Status Of File(s)	-		View the status of a file in Visual SourceSafe.
	Show Difference...		-		Opens the [Difference] window.
	Launch External Debugger...		-		Launches an external debugger tool..
	Launch Slave HEW...		-		Launches a slave HEW.

1.9 List of Window Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Function
Window	Cascade		-	-	Arrange all open windows so that they overlap.
	Tile Horizontally		-	-	Arrange all open windows horizontally.
	Tile Vertically		-	-	Arrange all open windows vertically.
	Arrange Icons		-	-	Line up all minimized windows.
	Close All		-	-	Close all open windows.
	Virtual desktop	Desktop Manager	-	-	Rename your configuration to a more meaningful name.
		Default 1-4	-	-	Switch desktop configurations

1.10 List of Help Menu

Menu	Menu Option		Keyboard Shortcuts	Toolbar Button	Function
Help	Help Topic		-	-	Opens the main High-performance Embedded Workshop help window.
	Technical Support	Create Bug Report...	-	-	Create a High-performance Embedded Workshop bug report.
		Check Website For Updates	-	-	Check for High-performance Embedded Workshop product updates or service packs.
	About High-performance Embedded Workshop...		-	-	Launches the About High-performance Embedded Workshop dialog box allowing the user to view the version of High-performance Embedded Workshop.
	Debugger Help		-	-	Shows the help window of the debugger when the debugger is connected.

2. List of Commands

2.1 Command List (Alphabetically Order)

Command Name	Short Name	Contents
!	-	Comment
ADD_FILE	AF	Adds a file to the current project
ASSERT	-	Checks if an expression is true or false
AUTO_COMPLETE	AC	Switches the auto-completion
BUILD *1	BU	Performs a build on the current project.
BUILD_ALL *1	BL	Performs a build all on the current project.
CACHE *2	-	Sets caching on or off
CHANGE_CONFIGURATION	CC	Sets the configuration to the specified configuration name
CHANGE_PROJECT	CP	Sets the specified project file as the current project
CHANGE_SESSION	CS	Sets the specified session as the current session
CLOSE_WORKSPACE	CW	Closes a workspace
DEFAULT_OBJECT_FORMAT	DO	Sets the object format to be used by default
ERASE	ER	Clears the Command Line window
EVALUATE	EV	Evaluates an expression
FILE_LOAD	FL	Loads an object (program) file
FILE_SAVE	FS	Saves memory to a file
FILE_UNLOAD	FU	Unloads an object file from memory
FILE_VERIFY *2	FV	Verifies file contents against memory
GENERATE_MAKE_FILE *1	GM	Generates a build makefile for the current workspace
GO	GO	Runs program
GO_RESET	GR	Runs program from reset
GO_TILL	GT	Runs program until specified addresses
HALT	HA	Halts program
HELP	HE	Displays help for Command Line or help on a command
INITIALIZE	IN	Initializes the debugging platform system
LOG	LO	Controls command output logging
MEMORY_COMPARE *2	MC	Compares memory contents
MEMORY_DISPLAY	MD	Displays memory contents
MEMORY_EDIT	ME	Modifies memory contents
MEMORY_FILL	MF	Fills a block of memory
MEMORY_FIND *2	MI	Finds a string in an area of memory
MEMORY_MOVE	MV	Moves a block of memory
MEMORY_TEST *2	MT	Tests a block of memory
OPEN_WORKSPACE	OW	Opens the specified workspace file
QUIT	QU	Exits HEW
RADIX	RA	Sets default input radix
REMOVE_FILE	REM	Removes a file from the current project
RESET	RE	Resets the microprocessor
SAVE_SESSION	SE	Saves the current session
SAVE_WORKSPACE	SW	Saves the current workspace
SLEEP	-	Delays command execution
STEP	ST	Steps through program (by instructions or source lines)
STEP_MODE	SM	Sets the step mode
STEP_OUT	SP	Steps out of the current function
STEP_OVER	SO	Steps through program without stepping into functions
STEP_RATE	SR	Sets rate of stepping
SUBMIT	SU	Executes a file of commands

TCL	-	Turns TCL commands on or off
TOOL_INFROMATION	TO	Outputs the tool information
UPDATE_ALL_DEPENDENCIES *1	UD	Updates the current projects build dependencies.

***1** : Available only when there is a toolchain installed.
***2** : Support for this command depends on the debugging platform.

For the syntax of each command, refer to the online help.

2.2 Command List (Listed as the Functions)

HEW Application Control Commands

Command Name	Short Name	Contents
ADD_FILE	AF	Adds a file to the current project
CHANGE_CONFIGURATION	CC	Sets the configuration to the specified configuration name
CHANGE_PROJECT	CP	Sets the specified project file as the current project
CHANGE_SESSION	CS	Sets the specified session as the current session
CLOSE_WORKSPACE	CW	Closes the specified workspace file
EVALUATE	EV	Evaluates an expression
OPEN_WORKSPACE	OW	Opens the specified workspace file
QUIT	QU	Exits HEW
RADIX	RA	Sets default input radix
REMOVE_FILE	REM	Removes a file from the current project
SAVE_SESSION	SE	Saves the current session
SAVE_WORKSPACE	SW	Saves the current workspace
TOOL_INFORMATION	TO	Outputs the tool information

Build Commands (Available only when there is a toolchain installed)

Command Name	Short Name	Contents
BUILD	BU	Performs a build on the current project
BUILD_ALL	BL	Performs a build all on the current project.
GENERATE_MAKE_FILE	GM	Generates a build makefile for the current workspace
UPDATE_ALL_DEPENDENCIES	UD	Updates the current projects build dependencies

Command Line Operating Commands

Command Name	Short Name	Contents
!	-	Comment
ASSERT	-	Checks if an expression is true or false
AUTO_COMPLETE	AC	Switches the auto-completion
ERASE	ER	Clears the Command Line window
HELP	HE	Displays help for Command Line or help on a command
LOG	LO	Controls command output logging
SLEEP	-	Delays command execution
SUBMIT	SU	Executes a file of commands
TCL	-	Turns TCL commands on or off

Execution Commands (Available when the debugger is connected)

Command Name	Short Name	Contents
GO	GO	Runs program
GO_RESET	GR	Runs program from reset
GO_TILL	GT	Runs program until specified addresses
HALT	HA	Halts program
INITIALIZE	IN	Initializes the debugging platform system
RESET	RE	Resets the microprocessor
STEP	ST	Steps through program (by instructions or source lines)
STEP_MODE	SM	Sets the step mode
STEP_OUT	SP	Steps out of the current function
STEP_OVER	SO	Steps through program without stepping into functions
STEP_RATE	SR	Sets rate of stepping

Memory Operation Commands (Available when the debugger is connected)

Command Name	Short Name	Contents
CACHE ***1**	-	Sets caching on or off
FILE_LOAD	FL	Loads an object (program) file
FILE_SAVE	FS	Saves memory to a file
FILE_UNLOAD	FU	Unloads an object file from memory
FILE_VERIFY ***1**	FV	Verifies file contents against memory
MEMORY_COMPARE ***1**	MC	Compares memory contents
MEMORY_DISPLAY	MD	Displays memory contents
MEMORY_EDIT	ME	Modifies memory contents
MEMORY_FILL	MF	Fills a block of memory
MEMORY_FIND ***1**	MI	Finds a string in an area of memory
MEMORY_MOVE	MV	Moves a block of memory
MEMORY_TEST ***1**	MT	Tests a block of memory

***1** : .Support for this command depends on the debugging platform.

For the syntax of each command, refer to the online help.

3. Regular Expressions

The HEW editor allows you to include special characters in search strings when performing a find, replace or find in files operation.

Find/Replace:

These characters are listed in the table below and explained underneath.

Character	Function
+	Matches one or more occurrences of the preceding item, except in a bracket expression. For example, a+ matches a, aa, aaa, and so on.
*	Matches zero or more occurrences of the preceding item, except in a bracket expression. For example, a* matches the empty string, a, aa, and so on.
?	Matches zero or one occurrence(s) of the preceding item, except in a bracket expression. For example, a? matches the empty string and a.
{ and }	Specify a cardinality range, formed as follows: {m,n}. This construct matches between m and n occurrences of the preceding item. For example, a{2,3} matches aa and aaa.This construct can also be formed using {m,} and {m}. The first matches m or more occurrences of the preceding item. For example, a{2,} matches aa, aaa, aaaa, and so on. The second matches exactly m occurrences of the preceding item. For example, a{2} matches aa.
[and]	Create a bracket expression. Bracket expressions create a set of items, any of which may be matched. For example, [abc] matches a, or b, or c. Within a bracket expression all regular expression special characters are treated as normal, non-special characters, except:- specifies a range of character values, based on their bit pattern. For example, [A-Za-z] matches all uppercase and lowercase English characters. To indicate - as a character in the bracket expression, it must be the first or last character in the set; for example, [-a-z] or [A-Z-].^ is special only when placed in the first character position within the bracket set. Using ^ in the first position complements the set of items to be matched. For example, [^a-z] matches all characters except for lowercase English letters. Finally, in order to include a] as a character in the bracket set, you must include it as the first character in the set, as in []abc] or [^]abc].
(and)	Group regular expression items into sub-expressions, which are treated as a single unit. For example, whereas ab* matches a, ab, abb, and so on, (ab)* matches the empty string, ab, abab, and so on. (and) are not treated as special characters inside a bracket expression.
\	Escapes a regular expression character, causing it to be treated as a regular character. For example, whereas (ab) indicates a sub-expression consisting of ab, \(ab\) denotes the sequence of characters (, a, b, and).Note: To specify the \ character in C++ source code, you must specify \\, as the C++ compiler treats the \ character as special, denoting the beginning of an escape sequence embedded in the C++ source code. In data files, or text controls in dialog boxes, however, the double backslash is not necessary.
^	Indicates that a regular expression or sub-expression is anchored at the beginning of the input string. For example, ^ab matches ab and abc, but not cab. Recall that ^ is treated differently in bracket expressions.
$	Indicates that a regular expression or sub-expression is anchored at the end of the input string. For example, ab$ matches ab and cab, but not abc.
|	Denotes alternation, or the creation of a set of equally valid, alternate expressions or sub-expressions, each of which can be matched. For example, ab|cd matches ab or cd.
.	Matches any code unit, except for those which indicate the logical end of a line.

Find in files:

These characters are listed in the table below and explained underneath.

Character	Function
?	This character matches any single character, except the newline character. For example, t?p matches “top”, “tip” but not “trap”.
*	This character matches any number of occurrences (0 or more) of any character except a newline. Thus, this character will not match across new lines. The * character will match as few occurrences as are necessary to make the rest of the pattern match. For example, t*o matches the “to” of “too”, the “tro” of “trowel” and the “ty o” of “sporty orange” but not “smart orange” because the * character does not match across a new line.
\n	This character matches the newline character.\n would be used to search for line endings or in patterns that cross line boundaries. Example 1: ;\n matches every occurrence of a newline following a semicolon Example 2: ;\nif searches for a semicolon, a new line and a line beginning with “if”.
\t	This character matches the tab character. Example 1: \t8 Finds every occurrence of a tab character followed by an 8. Example 2: init\t Finds every occurrence of a tab character following “init”.
[ ]	This matches any one character or a range of single characters listed within the brackets. Brackets cannot be nested. [-] specifies a range of characters e.g. [a-z] or [0-9]. The beginning character in the range must have a lower ASCII value than the ending character of the range. [~] matches a single character if it is not any one of the characters between [~ and]. This pattern also matches newline characters, unless the newline character is included within the brackets. Example 1: [AEIOU] Finds every uppercase vowel. Example 2: [<>?] Finds a literal <, > or ?. Example 3: [A-Za-z0-9_] Matches an upper or lowercase letter, a digit or an underscore. Example 4: [~0-9] Matches any character except a digit. Example 5: [\t\n] Matches a space, a tab or newline. Example 6: [\]] Matches a literal] if] is placed after \.
\	This is the regular expression override character. If the character following the backslash is a regular expression character, it is treated as a normal character. The backslash is ignored if it is followed by a normal (non-regular expression) character. Example 1: * Searches for every occurrence of an asterisk. Example 2: \\ Searches for every occurrence of a backslash.

4. Placeholders

This section describes how to use the placeholders, a feature provided by several of the High-performance Embedded Workshop components.

4.1 What is a Placeholder?

A placeholder is a special string, inserted into text, which is replaced at some subsequent time for the actual value. For example, one of the HEW placeholders is `$(FULLFILE)` which represents a file with a full path.

Suppose that you have an editor in `c:\myedit\myeditor.exe`, which can accept the file to be edited as a parameter. When invoking the editor (for example, you may want to open the file '`FILE1.C`' from the directory '`c:\files`'), the following shortcut could be made:

```
c:\myedit\myeditor.exe c:\files\FILE1.C
```

However, what happens if you want to open any file through this editor? The problem is that the command above is specific to '`c:\files\file1.c`'. What we want to be able to do is to tell the HEW to use the editor specified but to open the file that we have chosen at that time. To do this, you can substitute the specific name of the file for a general **Placeholder**:

```
c:\myedit\myeditor.exe $(FULLFILE)
```

Now whenever the HEW launches the editor with a file, it knows that it has to replace the `$(FULLFILE)` placeholder with the file you have selected.

4.2 Inserting a Placeholder

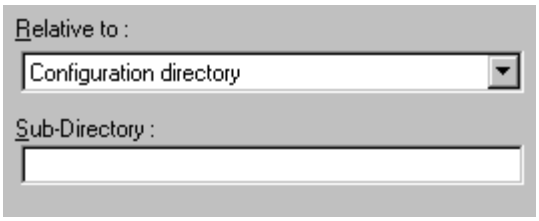
To insert a placeholder, select in any of the following operations.

Example 1:

1. Place the insertion cursor at the point you would like to insert the placeholder.
2. Click the placeholder button. A pop-up menu will be displayed which lists all of the placeholders that are valid for the associated edit box.
3. Select the desired placeholder from the pop-up menu. The equivalent placeholder will be inserted into the edit box.

Example 2:

1. Select the required placeholder other than [Custom directory] from the drop-down list box and specify a sub-directory relative to the directory shown by the placeholder.
2. If you select [Custom directory], specify an absolute directory path in the [Sub-Directory] field.



Example 3:

1. Place the insertion cursor at the point you would like to insert the placeholder
2. Select the required placeholder from the drop-down list box.
3. Click the [Insert] button.

Example 4:

1. Alternatively, if you know the placeholder already, type it into the field directly. Ensure that you type the placeholder name in uppercase and that it is preceded by '`$(`' and followed by '`)`'.

   This is correct:

   ```
   $(FILEDIR)
   ```

   These are incorrect:

   ```
   $(Filedir)
   $( FILEDIR )
   $FILEDIR
   ```

4.3 Available Placeholders

The table below lists the available placeholders and their meanings, along with an example of their use.

Placeholder	Meaning	Expanded placeholder (example)
$(FULLFILE)	Filename (including full path)	c:\hew\workspace\project\file.src
$(FILEDIR)	File directory	c:\hew\workspace\project
$(FILENAME)	Filename (excluding path, including extension)	file.src
$(FILELEAF)	Filename (excluding path and extension)	file
$(EXTENSION)	File extension	src
$(WORKSPDIR)	Workspace directory	c:\hew\workspace
$(WORKSPNAME)	Workspace name	workspace
$(PROJDIR)	Project directory	c:\hew\workspace\project
$(PROJECTNAME)	Project name	project
$(CONFIGDIR)	Configuration directory	c:\hew\workspace\project\debug
$(CONFIGNAME)	Configuration name	debug
$(HEWDIR)	HEW installation directory	c:\hew
$(TCINSTALL)	Toolchain install directory (on option dialog)	c:\hew\toolchains\renesas\sh\511
$(TOOLDIR)	Tool installation directory (Tools Administration)	c:\hew\toolchains\renesas\sh\511
$(TEMPDIR)	Temp directory	c:\temp
$(WINDIR)	Windows® directory	c:\windows
$(WINSYSDIR)	Windows® system directory	c:\windows\system
$(EXEDIR)	Command directory	v:\vc\win32
$(USERNAME)	User login (Version control)	JHARK
$(PASSWORD)	User password (Version control)	214436
$(VCDIR)	“Virtual” version control directory	“c:\project” is mapped to “x:\vc\project”
$(COMMENT)	Comment (Version control)	“Please Enter Comment” dialog is invoked
$(LINE)	Line number of an error/warning	12

In the table above, we are assuming that:

- a file path is “c:\hew\workspace\project\file.src”
- a workspace named “workspace” is located at “c:\hew\workspace”
- a project named “project” is located at “c:\hew\workspace\project”
- a configuration named “debug” has a configuration directory located at “c:\hew\workspace\project\debug”
- HEW.EXE is installed in “c:\hew”
- a *.HRF file of a toolchain (i.e. compiler, assembler, linker) is located in “c:\hew\toolchain\renesas\sh\511”. This is referred to as $(TCINSTALL) on the options setting dialogs of the [Build] menu and as $(TOOLDIR) on the [Tools Administration] dialog
- the Windows® operating system is installed in “c:\windows” and the Windows® system directory is located at “c:\windows\system”
- a version control executable path is “v:\vc\win32\ss.exe”; a user name and its password to login to the version control system are “JHARK” and “214436” respectively; $(COMMENT) is specified in a command line to the version control executable; “c:\project” is mapped to “x:\vc\project” on the [Projects] tab of the [Version Control Setup] dialog, which is invoked via [Tools->Version Control->Configure…].
- an error of compiler or assembler occurred at line 12

Note:

Not all of the placeholders are relevant in every field. For example, the $(LINE) placeholder has no meaning when specifying a dependent file's location. $(USERNAME), $(PASSWORD), $(VCDIR), and $(COMMENT) placeholders are acceptable only in version control. If you enter a placeholder into an edit field where it is not acceptable you will be informed.

4.4 Placeholder Tips

Placeholders are there to allow you to create flexible paths to the various files used by the system.

- If there is a placeholder pop-up menu  next to an edit field into which you are about to enter a path or file, you should consider how you can use a placeholder to make that path or file definition flexible.
- If you use several configurations, then the `$(CONFIGDIR)` placeholder is very useful to ensure that files can be written to and from the current configuration's directory.
- Wherever possible, use a placeholder. They can always be removed or added later so don't be afraid to experiment.

5. I/O File Format

HEW formats the [IO] window based on information it finds in an I/O Register definition file. When you select a debugging platform, HEW will look for a "<*device*>.IO" file corresponding to the selected device and load it if it exists. This file is a formatted text file that describes the I/O modules and the address and size of their registers. You can edit this file, with a text editor, to add support for memory mapped registers or peripherals you may have specific to your application e.g. registers in an ASIC device mapped into the microcomputer's address space.

File format

Each module name must be defined in the [Modules] definition section and the numbering of each module must be sequential. Each module corresponds to a register definition section and within the section each entry defines an I/O register.

The [BaseAddress] definition is for devices where the location of I/O registers moves in the address space depending on the CPU mode. In this case, the [BaseAddress] value is the base address of the I/O registers in one specific mode and the addresses used in the register definitions are the address locations of the registers in the same mode. When the I/O register file is actually used, the [BaseAddress] value is subtracted from the defined register address and the resultant offset added to the relevant base address for the selected mode.

Each module has a section that defines the registers forming it along with an optional dependency, the dependency is checked to see if the module is enabled or not. Each register name must be defined in the section and the numbering of each register must be sequential. The dependency is entered in the section as dep=<reg> <bit> <value>.

1. <reg> is the register id of the dependency.
2. <bit> is the bit position within the register.
3. <value> is the value that the bit must be for the module to be enabled.

The [Register] definition entry is entered in the format id=<name> <address> [<size> [<absolute>[<format>[<bitfields>]]]].

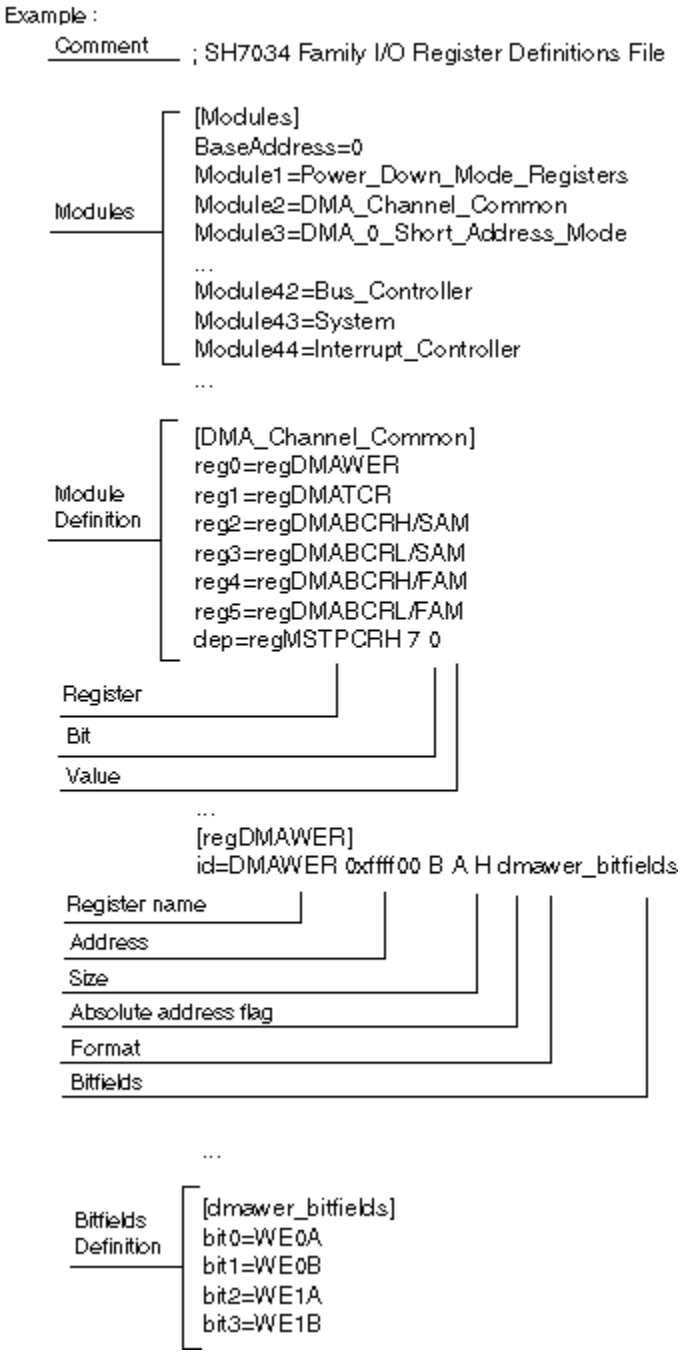
1. <name> register name to be displayed.
2. <address> address of the register.
3. <size> which may be B, W or L for byte, word, or long word (default is byte).
4. <absolute> which can be set to A if the register is at an absolute address. This is only relevant if the I/O area address range moves about on the CPU in different modes. In this case, if a register is defined as absolute the base address offset calculation is not performed and the specified address is used directly.
5. <format> Format for register output. Valid values are H for Hexadecimal, D for decimal, and B for binary.
6. <bitfields> section defining the bits within the register.

Bitfield sections define the bits within a register each entry is of the type bit<no>=<name>.

1. <no> is the bit number.
2. <name> is a symbolic name of the bit.

Comment lines are allowed and must start with a ";" character.

Example :


Comment
; SH7034 Family I/O Register Definitions File
Modules
[Modules]
BaseAddress=0
Module1=Power_Down_Mode_Registers
Module2=DMA_Channel_Common
Module3=DMA_0_Short_Address_Mode
...
Module42=Bus_Controller
Module43=System
Module44=Interrupt_Controller
...
Module Definition
[DMA_Channel_Common]
reg0=regDMAWER
reg1=regDMATCR
reg2=regDMABCRH/SAM
reg3=regDMABCRL/SAM
reg4=regDMABCRH/FAM
reg5=regDMABCRL/FAM
dep=regMSTPCRH 7 0
Register
Bit
Value
...
[regDMAWER]
id=DMAWER 0xffff00 B A H dmawer_bitfields
Register name
Address
Size
Absolute address flag
Format
Bitfields
...
Bitfields Definition
[dmawer_bitfields]
bit0=WE0A
bit1=WE0B
bit2=WE1A
bit3=WE1B

6. Symbol File Format

In order for HEW to be able to understand and decode the symbol file correctly, the file must be formatted as a Pentica-B file:

1. The file must be a plain ASCII text file.
2. The file must start with the word “BEGIN”.
3. Each symbol must be on a separate line with the value first, in hexadecimal terminated by an “H”, followed by a space then the symbol text.
4. The file must end with the word “END”.

Example:

BEGIN

11FAH Symbol_name_1

11FCH Symbol_name_2

11FEH Symbol_name_3

1200H Symbol_name_4

END

Note:

Support for this function depends on the debugging platform.

7. Keyboard Shortcuts

All major commands in the HEW application can be driven by the keyboard. Below is a list of all keyboard commands in the application.

Function key	Key	Function
CTRL	0-9	Reserved for use for template insertion.
ALT	K	Open workspace window.
ALT	U	Open output window.
ALT	F4	Exit the application.
ALT	BACKSPACE	Undo (alternative in the editor to CTRL+Z)
CTRL	A	Select all in the editor. May also work in other windows.
CTRL	C	Copy.
CTRL	D	Open the disassembly window.
CTRL	F	Find.
CTRL	G	Goto line.
CTRL	H	Replace.
CTRL	I	Open the IO window.
CTRL	L	Open the command line window.
CTRL	K	Open the stack trace window.
CTRL	M	Open the memory window.
CTRL	N	Create a new source file in the editor.
CTRL	O	Open the open file dialog.
CTRL	P	Print.
CTRL	R	Open the registers window.
CTRL	S	Save the current file.
CTRL	U	Open the status window.
CTRL	V	Paste.
CTRL	X	Cut.
CTRL	Y	Redo
CTRL	Z	Undo
CTRL	Break	Stop build.
CTRL	F2	Toggle bookmark.
CTRL	F4	Close file.
CTRL	F7	Build file.
CTRL	F9	Enable and disable breakpoints
CTRL	Cursor up	Scroll window up and leave the cursor in the same place in the editor.
CTRL	Cursor down	Scroll window down and leave the cursor in the same place in the editor.
CTRL	Cursor left	Move the cursor to the previous word in the editor.
CTRL	Cursor right	Move the cursor to the next word in the editor.
CTRL	BACKSPACE	Delete current word.
CTRL	END	Sends the cursor in the editor to the end of the current file.
CTRL	HOME	Returns the cursor in the editor to the beginning of the current file.
CTRL	TAB	Switch windows in the editor.
CTRL	F6	Switch windows in the editor.
SHIFT	F2	Previous bookmark.
SHIFT	F5	Reset Go.
SHIFT	ESC	Hide output window.
SHIFT	F11	Step out.
SHIFT	F10	Show context menu. Same as right-click pop-up menu.
SHIFT	F12	Refresh all windows.
SHIFT	DELETE	Editor cut operation. (alternative in the editor to CTRL+X)
SHIFT	Cursor up.	Select the next line up.
SHIFT	Cursor down.	Select the next line down
SHIFT	HOME	Selects from the cursor to the beginning of the current line in the editor.
SHIFT	END	Selects from the cursor to the end of the current line in the editor.
SHIFT	TAB	Move the tab back in the editor.
SHIFT+CTRL	8	Show white space characters in the editor.

SHIFT+CTRL	G	Open image window.
SHIFT+CTRL	K	Open TCL toolkit.
SHIFT+CTRL	L	Line deletion in editor.
SHIFT+CTRL	M	Match braces.
SHIFT+CTRL	S	Save all.
SHIFT+CTRL	T	Insert template.
SHIFT+CTRL	V	Open waveform window.
SHIFT+CTRL	W	Select a word in the editor.
SHIFT+CTRL	END	Selects from the cursor to the end of the file.
SHIFT+CTRL	HOME	Selects from the cursor to the beginning of the file.
SHIFT+CTRL	Cursor left	Selects the previous word in the editor.
SHIFT+CTRL	Cursor right	Selects the next word in the editor.
None	DEL	Clear.
None	ESC	Halt
None	F1	Context sensitive help.
None	F2	Next bookmark.
None	F3	Find Next.
None	F4	Find in files.
None	F5	Go.
None	F7	Build
None	F10	Step over.
None	F11	Step in.
None	F12	Refresh window.
None	Cursor up	Move cursor up in the editor.
None	Cursor down	Move cursor down in the editor.
None	Cursor left	Move cursor left in the editor.
None	Cursor right	Move cursor right in the editor.
None	End	Move cursor to the end of the current line in the editor.
None	Home	Move the cursor to the beginning of the current line in the editor.
None	Insert	Toggle insert and overwrite mode.
None	Page up	Moves the page in the editor up.
None	Page down	Moves the page in the editor down.
None	Return	Carriage return in the editor.
None	TAB	Insert a tab in the editor.
None	ALT + Mouse select	Column selection in the editor.
CTRL+ALT	Page up	Move to next tab. For example output window.
CTRL+ALT	Page down	Move to previous tab. For example output window.
CTRL+ALT	Tab	Move to next window.
ALT+SHIFT	BACKSPACE	Redo (alternative in the editor to CTRL+Y)

Note:

Support for this function depends on the debugging platform.

8. Drag and Drop in the HEW Debugger

When using the HEW debugger it is possible for each debug component to interact with the others. This can be achieved simply by dragging objects from one view to another.

Some examples are listed below:

1. It is possible to drag a label from the labels view onto other debug views. So for example if you drag a label onto the disassembly window it will scroll to the address that the label is located at.
2. It is possible to drag a watch variable from the editor into the watch window. This automatically adds the watch variable to the window.
3. Dragging a function name from the editor into the disassembly should jump the disassembly view to the label location.

9. Using Labels to View Your Code

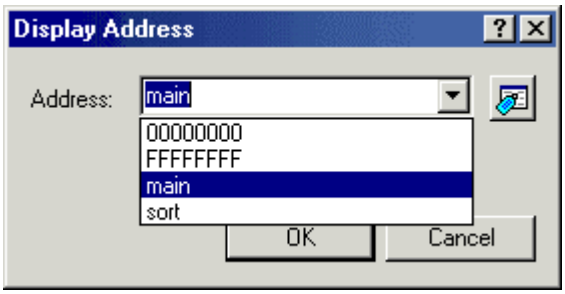
Labels are a useful way of navigating through your debug module. It is possible to use labels in any edit field that allows addresses. If you enter a label in such a field then the built in evaluator will check the label and then convert it to an address. This allows you to enter evaluations such as "main+100" or "_MyFunction+100".

This means that any times that labels are used the addresses which will be evaluated are not fixed. This is especially useful if you are using a command line batch file to set a number of breakpoints. The command line batch file might always need to set a breakpoint on a certain function and this can be achieved by using a label.

Using the label allows the code to change without affecting the batch file contents.

HEW 3.1 onwards also supports a number of easy ways to use label completion. Each instance of the address edit field will have the labels button  attached to it. This allows you to click and browse the current labels list.

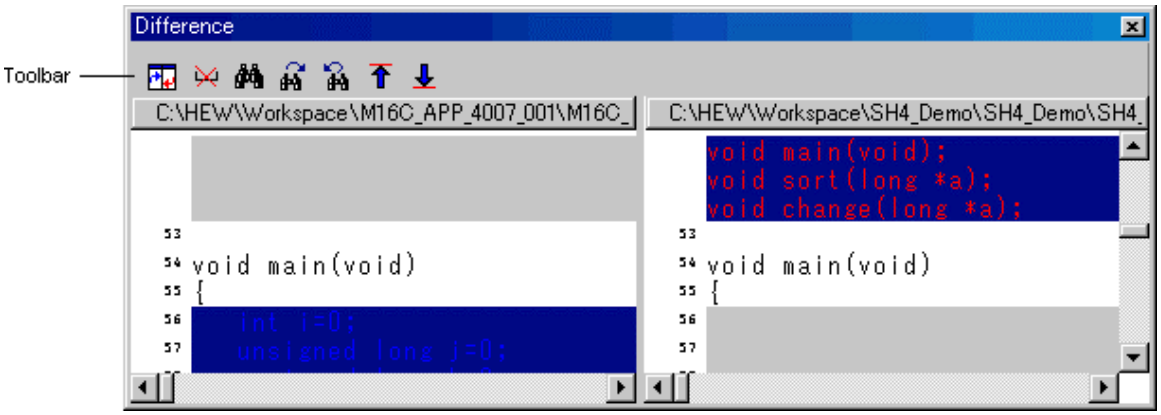
The dialog below also shows an example of a label pick list. This stores the last twenty entries made into address fields throughout the entire HEW application. This means if you are entering a label multiple times it should be much faster and efficient if you use this recently used address field list. This control is available for all instances of the address edit field where the input is evaluated.



10. Integrated Toolbars in a Components View

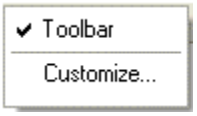
The HEW 4.0 onwards has the capability to include a toolbar in a views client area. This toolbar allows the views functionality to be accessed quickly from this integrated toolbar.

Various views in the HEW system have this functionality. One example is the Difference view. This is shown below:

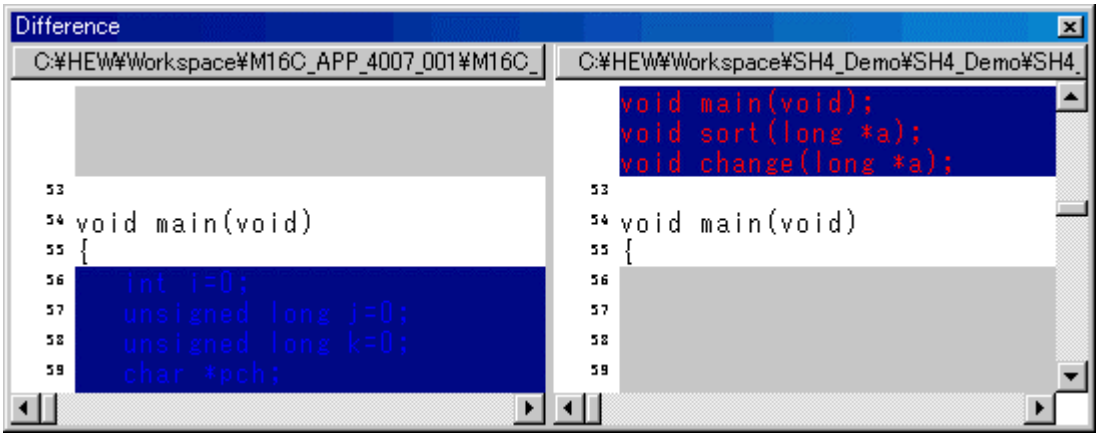


The toolbar allows access to key Difference view features.

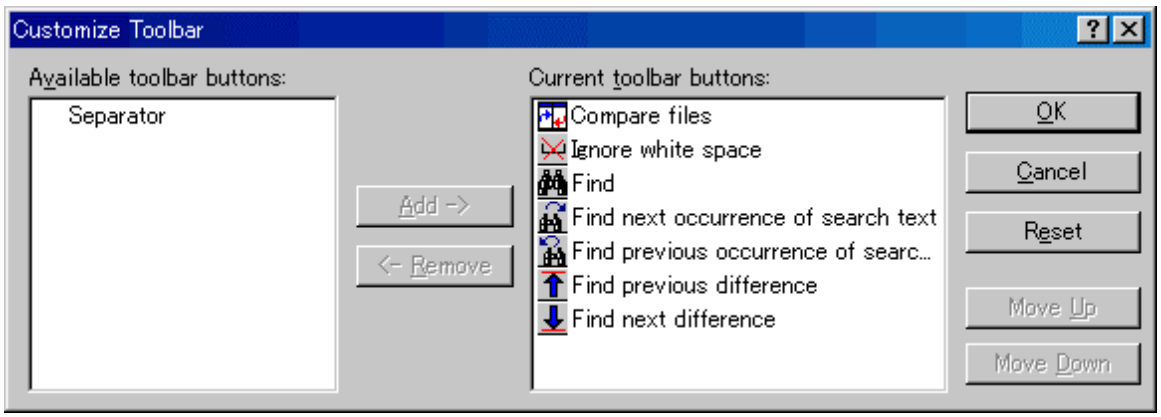
It is also possible to customize the toolbar further. This can be achieved via the pop-up menu of the toolbar or the component itself. If you right click on the toolbar itself the following menu is displayed:



The top menu option named "Toolbar" switches showing/hiding of the toolbar (the toolbar is hidden in the figure below).



The final menu option named "Customize..." launches the [Customize Toolbar] dialog. This is shown below:



This dialog allows you to modify the displayed buttons and change the ordering. The list on the left named [Available toolbar buttons] displays all toolbar buttons not currently in use on the toolbar. The list named [Current toolbar buttons] displays all of the toolbars currently added to the components' toolbar.

To add the currently selected buttons to a toolbar:

1. Select the toolbar button you wish to add from the [Available toolbar buttons] list.
2. Press the [Add] button. The toolbar is added to the bottom of the list.
3. Click [OK].

To move the currently selected buttons:

1. Select the toolbar you wish to move in the [Current toolbar buttons] list.
2. Click [Move up] or [Move down] until it is in the desired position.
3. Click [OK].

To remove the currently selected buttons from a toolbar:

1. Select the toolbar button you wish to remove from the [Current toolbar buttons] list.
2. Press the [Remove] button. The toolbar is added to the [Available toolbar buttons] list.
3. Click [OK].

11. To Build in Toolchain for HEW1.x

When a project created in HEW1.x is used without upgrading to new toolchain that has been registered in HEW2.x onwards, the toolchain for the old version must be registered. Select the 'hrf' file for the old toolchain with the 'Register...' button by selecting [Tools–>Administration...]. Build can be executed on HEW2.x onwards by using the old toolchain.

However, note that a workspace which has been opened in HEW2.x onwards cannot be opened in HEW1.x.

In HEW2.x onwards , a new project for the old version cannot be created. When a project for the old version in HEW1.x is created, use HEW1.x.

12. HMAKE User Guide

12.1 Command Line

The following section describes the command line that should be used to execute the hmake program on a file using none or more of the available options.

Basic structure

The command line must be of the following syntax:

hmake <make file you wish to execute> <parameter list>

If a file is specified without an extension then ".mak" will be appended to it. The parameter list may include none or more of the parameters listed in the following section. The parameters list may appear before the make file name if you wish. Each parameter must be separated by at least one white space character. Parameters are not case sensitive. If no parameters are given and no file is given then help information will be displayed.

Exit codes

If there are any syntax errors in the make file being executed or if any process executed whilst running the make file returns an invalid error code then hmake will exit with code 1. Otherwise hmake will exit with code 0 (See below for file syntax and how to specify exit code conditions).

Parameters

The following table shows the available parameters and their function:

Parameter	Function
/A	Execute all commands regardless of input/output file status. Equivalent to a Build All.
/N	Use status of input/output files to calculate what commands need to be executed (as normal) and then display the commands but do not execute them.
/?	Display help info.

12.2 File Syntax

There are four basic types of statement used in a hmake file, the variable declaration the description block, the comment and the message command. These can be combined in any order to produce a hmake file but a variable must be declared in a variable declaration before it is used in a description block or other variable declaration. The first "all" statement used in nmake files is not required in a hmake file. Commands are executed in order, as they appear in the make file.

Note: the "→" character is used to show were a tab character must be used in order to keep the make file syntactically correct.

Variable declarations

A variable declaration declares a variable which can then be used in any statement throughout the rest of the hmake file. A declaration has the following syntax:

<variable name> = <value>

Any number of white space characters are allowed between the variable name and the '=' sign and the value and the '=' sign. The value may be split over several lines using a '\' character. If the value contains '\' characters within the main text then these are taken literally. Only '\' characters followed by a new line are considered to indicate a value wrapping over more than one line.

There follows some examples of valid variable declarations:

```
EXECUTABLE = c:\dir\prog.exe
OUTPUT = c:\dir2\file1.out
INPUT = c:\dir2\file1.c
DEPEND = c:\dir2\file2.h \
    c:\dir2\file3.h \
    c:\dir2\file4.h
```

In order to use a variable later in the hmake file write the variable name with "$(" added to the front and ")" added to the back. The variable name (along with the "$()" characters) will be substituted with the variables value. For examples of this see later under description blocks. Only alphanumeric characters and underscore characters are allowed in variable names. It is possible to use a variable inside the declaration of a different variable but all variables must be declared before they are used.

12.3 Description blocks

Basic outline

A description block specifies one or more targets, zero or more dependants and a list of commands which should be executed if the newest dependent is newer than the newest target. If none of the targets exist and/or none of the dependants exist then the commands will always be executed. It is not necessary to specify any dependants if you wish the commands to always be executed. A description block has the following syntax:

```
<target1> <target2> … : <dependant1> <dependant2> …
→ <command1>
→ <command2>
→ …
→ <commandn>
```

Any number of white space characters are allowed between the last target and the ':' character and the first dependant and the ':' character. No white space is allowed before the first target. Each target and each dependant must be separated by at least one white space character. A tab character must be present at the start of a line containing a command. Variables may be used in a description block using the syntax specified above under variable declarations.

There follows some examples of valid description blocks (one of which uses the variable specified above under variable declarations):

```
c:\dir1\file1.obj : c:\dir1\file1.c c:\dir1\file1.h
→ gcc c:\dir1\file1.c
$(OUTPUT) : $(INPUT) $(DEPEND)
→ $(EXECUTABLE) $(INPUT)
```

Special commands

There are two special commands which can be used in a description block. The "cd" command changes the current directory and the "set" command sets an environment variable which will then be in use for the duration of the make file execution. Both are used in the same way as the DOS equivalents.

There follows some examples of valid description blocks which use these commands:

```
CHANGEDIR :
→ cd c:\dir1\dir2
SETENV:
→ set VAR1=value1
→ set VAR2=value2
→ set VAR3=value3
```

It does not matter that CHANGEDIR and SETENV are not file names. They will be treated as files that do not exist and so the commands will always be executed.

Sub command files

If you wish hmake to generate a sub command file for you then the command part of the description block should be specified as follows (this replaces <commandn> above):

```
→ <command start> <<
→ <sub command1>,
→ <sub command2>,
→ …
→ <sub commandn>
<<<command end>
```

This will generate a sub command file, in the windows temporary directory, which will contain the lines <sub command1>, <sub command2> etc. This command file will be deleted once the make process has completed. The name of the command file will be substituted for all the text between the two “<<”’s. You do not have to worry about the name of the sub command file. This is generated by hmake.

For example:

```
c:\dir1\file1.obj : c:\dir1\file1.c c:\dir1\file1.h
→ gcc @”<<
→ -c -o c:\dir1\file1.obj c:\dir1\file1.c
<<”
```

If the sub command file generated has the name “c:\temp\hmk111.cmd” then the following would be executed by hmake (assuming c:\dir1\file1.obj is out of date):

```
gcc @”c:\temp\hmk111.cmd”
```

The command file (c:\temp\hmk111.cmd) would contain:

```
-c -o c:\dir1\file1.obj c:\dir1\file1.c
```

It is possible to include more than one command in the description block and to use combinations of the standard, and sub command file commands.

12.4 Comments

A '#' character signifies a comment. When this character appears as the first character on a line the rest of the line (up until the next new line character) is ignored. There follows examples of valid comments:

```
# My hmake file
# Variable declaration
OUTPUT= c:\dir1\file1.obj
# Descriptor
$(OUTPUT) : c:\dir1\file1.c c:\dir1\file1.h
→ set VAR1=value1
→ gcc c:\dir1\file1.c
```

A comment must occupy its own line in the hmake file. It is not possible to put comments on the end of other statements.

12.5 Message commands

The message command is used to output a line of text to standard out whilst a make file is executing. These text lines will be output in the order they appear in the make file, in amongst output from any executables being executed as appropriate. No buffering of output text will take place. A message command has the following syntax:

```
!MESSAGE <text to output>
```

A new line character is assumed to come after the last character in <text to output>. Any white space between !MESSAGE and <text to output> will be ignored. There follows an example of a valid message command:

```
!MESSAGE Executing C Compiler
```

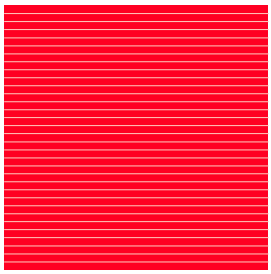
High-performance Embedded Workshop V.4.00
User's Manual

Publication Date: Jan. 19, 2005 Rev.1.00

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

Edited by: Microcomputer Tool Development Department
Renesas Solutions Corp.

High-performance Embedded Workshop V.4.00
User's Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan